



**M** 2015

# **SISTEMA ACÚSTICO-ÓPTICO DE SEGUIMENTO DE ALVO SUBMERSO**

**FÁBIO FERNANDO FERREIRA DA SILVA**  
DISSERTAÇÃO DE MESTRADO APRESENTADA  
À FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO EM  
ÁREA CIENTÍFICA

**Faculdade de Engenharia da Universidade do Porto**



# **Sistema Acústico-Óptico de Seguimento de Alvo Submerso**

**Fábio Fernando Ferreira da Silva**

Dissertação realizada no âmbito do  
Mestrado Integrado em Engenharia Eletrotécnica e de Computadores  
Major de Automação

Orientador: Aníbal Matos

Coorientador: Nuno Cruz

20 de Janeiro de 2015



A Dissertação intitulada

“Sistema Acústico-Óptico de Seguimento de Alvo Submerso”

foi aprovada em provas realizadas em 19-02-2015

o júri

*Paulo José Cerqueira Gomes da Costa*

Presidente Professor Doutor Paulo José Cerqueira Gomes da Costa  
Professor Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores  
da Faculdade de Engenharia da Universidade do Porto

*Marcos Silva Martins*

Professor Doutor Marcos Silva Martins  
Professor Auxiliar Convidado do Departamento Eletrónica Industrial da Universidade  
do Minho

*Aníbal Castilho Coimbra de Matos*

Professor Doutor Aníbal Castilho Coimbra de Matos  
Professor Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores  
da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.

*Fábio Fernando Ferreira da Silva*

Autor - Fábio Fernando Ferreira da Silva

Faculdade de Engenharia da Universidade do Porto

# Resumo

O *Ocean Systems Group*, um grupo associado à Unidade de Robótica e Sistemas Inteligentes do INESC TEC, está inserido em vários projetos de robótica aquática. Este grupo tem grande interesse em que Veículos Autónomos de Superfície (ASVs) sejam capazes de efetuar missões de seguimento.

Esta dissertação centra-se na aquisição e fusão de dados provenientes de dois sistemas, com o intuito de melhorar o seguimento, ou seja, obter informações de posicionamento com mais exatidão. A inovação aqui presente está na integração de um sistema que começa cada vez mais a ser utilizado na robótica, a visão computacional.

A proposta de solução envolve a conciliação entre ambos os sistemas, visão/óptico e o acústico, de modo a beneficiar das suas vantagens.

O trabalho será concluído com a apresentação de resultados experimentais que permitam avaliar o desempenho da solução proposta.



# Abstract

Ocean Systems Group is an associate group of INESC TEC's Centre for Robotics and Intelligent Systems which research activities are directed towards marine robotics. This group is highly interested in working with Autonomous Surface Vehicles (ASVs) in order to use them in tracking missions.

This thesis is based on data collection and processing from two systems in order to improve tracking and therefore acquire a more accurate positioning. The new concept introduced by this thesis is the incorporation of a system which has been growing in the robotics field: computer vision.

The suggested solution involves working with both vision/optical and acoustic systems so as to benefit from its advantages.

This work will conclude with the presentation of experimental results which will make possible to evaluate the solution's performance.





# Agradecimentos

É com muita satisfação que exprimo aqui o mais profundo agradecimento a todas as pessoas que ajudaram na realização deste trabalho.

Em primeiro lugar gostaria de agradecer ao Professor Aníbal Matos, orientador desta dissertação, pelo apoio e disponibilidade apresentados.

Uma palavra de gratidão aos colegas da OceanSys e ao meu coorientador Professor Nuno Cruz pela ajuda e disponibilidade.

Aos meus amigos pela amizade e companheirismo.

Por último, um agradecimento especial aos meus pais e irmãos por estarem sempre presentes e à minha namorada pela paciência, dedicação e incentivo durante todo este trabalho.

Fábio Silva



*“Logic will get you from A to Z, imagination will get you everywhere.”*

Albert Einstein



# Conteúdo

|          |                                     |           |
|----------|-------------------------------------|-----------|
| <b>1</b> | <b>Introdução.....</b>              | <b>1</b>  |
| 1.1      | Motivação.....                      | 1         |
| 1.2      | Objetivos .....                     | 1         |
| 1.3      | Estrutura do Documento .....        | 2         |
| <b>2</b> | <b>Estado da Arte.....</b>          | <b>4</b>  |
| 2.1      | Modelo do sistema .....             | 4         |
| 2.2      | Método de localização acústica..... | 7         |
| 2.2.1    | Método LBL.....                     | 7         |
| 2.3      | Método de localização óptico.....   | 8         |
| 2.3.1    | Projeção da imagem .....            | 9         |
| 2.3.2    | Pré-processamento .....             | 12        |
| 2.3.3    | Segmentação .....                   | 13        |
| 2.3.4    | Análise e representação.....        | 17        |
| 2.3.5    | Reconhecimento e classificação..... | 18        |
| 2.4      | Filtro de Kalman .....              | 19        |
| 2.5      | Filtro de Kalman Estendido .....    | 21        |
| <b>3</b> | <b>Plataformas.....</b>             | <b>25</b> |
| 3.1      | ASV Zarco .....                     | 25        |
| 3.2      | AUV MARES .....                     | 26        |
| 3.3      | Boias.....                          | 26        |
| 3.4      | Câmara subaquática .....            | 27        |
| 3.5      | Modelo de Comunicação.....          | 28        |
| 3.6      | Repositório .....                   | 29        |
| <b>4</b> | <b>Trabalho desenvolvido.....</b>   | <b>31</b> |
| 4.1      | Calibração .....                    | 31        |
| 4.2      | Alvo.....                           | 34        |
| 4.2.1    | Escolha de propriedades.....        | 34        |
| 4.2.2    | Testes .....                        | 36        |

|          |  |           |
|----------|--|-----------|
| 4.2.3    | Solução final.....                       | 36        |
| 4.3      | Algoritmos desenvolvidos.....            | 37        |
| 4.3.1    | Visão computacional.....                 | 37        |
| 4.3.2    | Acústica.....                            | 45        |
| 4.3.3    | Fusão e estimação .....                  | 47        |
| <b>5</b> | <b>Implementação .....</b>               | <b>52</b> |
| <b>6</b> | <b>Resultados .....</b>                  | <b>58</b> |
| 6.1      | Ambiente e ferramentas .....             | 58        |
| 6.2      | Teste nº1 .....                          | 59        |
| 6.3      | Teste nº2.....                           | 62        |
| <b>7</b> | <b>Conclusões e Trabalho Futuro.....</b> | <b>67</b> |
| 7.1      | Conclusão.....                           | 67        |
| 7.2      | Trabalho Futuro.....                     | 68        |
| <b>A</b> | <b>Anexos .....</b>                      | <b>69</b> |
| A.1      | Fluxogramas .....                        | 69        |
| A.2      | Máquinas de Estados.....                 | 71        |
|          | <b>Referências.....</b>                  | <b>72</b> |

# Lista de Figuras

|      |  |    |
|------|--|----|
| 2.1  | Modelo genérico de um sistema baseado em visão. ....               | 9  |
| 2.2  | Formação de imagem. ....   | 9  |
| 2.3  | Diagrama de transformação. ....                                    | 12 |
| 2.4  | Processo de transformação de uma operação local. ....              | 12 |
| 2.5  | Exemplo de uma operação local. ....                                | 13 |
| 2.6  | Sistema genérico de aplicação do Filtro de Kalman. ....            | 19 |
| 2.7  | Ciclo de processamento do Filtro de Kalman. ....                   | 20 |
| 2.8  | Matrizes Jacobianas. ....  | 22 |
| 2.9  | Ciclo de processamento do Filtro de Kalman Estendido. ....         | 23 |
| 3.1  | ASV Zarco. ....  | 25 |
| 3.2  | AUV MARES. ....  | 26 |
| 3.3  | Boia acústica. ....  | 27 |
| 3.4  | Câmara subaquática. ....   | 28 |
| 4.1  | Imagem captada para calibração. ....                               | 32 |
| 4.2  | Imagem com a seleção dos cantos. ....                              | 32 |
| 4.3  | Imagem com os quadrados selecionados. ....                         | 33 |
| 4.4  | Alvo inicial. ....   | 35 |
| 4.5  | Alvo final. ....   | 36 |
| 4.6  | Resultado do perímetro sem alvo. ....                              | 38 |
| 4.7  | Resultado do perímetro com alvo. ....                              | 38 |
| 4.8  | Segmentação de imagem com limiar a 110. ....                       | 40 |
| 4.9  | Segmentação de imagem com limiar a 135. ....                       | 40 |
| 4.10 | Estimação da profundidade com alvo a 1.65 m. ....                  | 42 |
| 4.11 | Estimação da profundidade com alvo a 0.8 m. ....                   | 42 |
| 4.12 | Ilustração da obtenção da orientação. ....                         | 43 |
| 4.13 | Ilustração da mudança de origem. ....                              | 44 |
| 4.14 | Resultado final do sistema óptico. ....                            | 44 |
| 4.15 | Esquema de troca de sinais acústicos ....                          | 45 |
| 4.16 | Ilustração do cálculo da posição. ....                             | 47 |
| 4.17 | Ilustração da fusão de dados. ....                                 | 48 |
| 5.1  | Diagrama de interação entre processos. ....                        | 53 |
| 5.2  | Excerto do código para comunicação entre clientes e servidor. .... | 53 |
| 5.3  | Formato da string. ....  | 54 |
| 5.4  | Excerto de código para registo no centralizador. ....              | 54 |

|      |   |    |
|------|---|----|
| 5.5  | Excerto de código do temporizador. ....                 | 55 |
| 6.1  | Ambiente de testes. ....                                | 59 |
| 6.2  | Resultado do subsistema óptico para a posição 1. ....   | 60 |
| 6.3  | Resultado do subsistema acústico para a posição 1. .... | 60 |
| 6.4  | Resultado do subsistema de visão para a posição 2. .... | 61 |
| 6.5  | Resultado do subsistema acústico para a posição 2. .... | 61 |
| 6.6  | Esquema do teste realizado. ....                        | 62 |
| 6.7  | Foto da realização do teste. ....                       | 62 |
| 6.8  | Posição obtida em x. ....                               | 63 |
| 6.9  | Posição obtida em y. ....                               | 63 |
| 6.10 | Velocidade estimada pelo EKF em x. ....                 | 64 |
| 6.11 | Velocidade estimada pelo EKF em y. ....                 | 64 |
| 6.12 | Velocidade estimada pelo RLS em x. ....                 | 65 |
| 6.13 | Velocidade estimada pelo RLS em y. ....                 | 65 |
| A.1  | Processo Óptico. ....                                   | 69 |
| A.2  | Processo Acústico. ....                                 | 70 |
| A.3  | Processo de Fusão e Estimação. ....                     | 70 |
| A.4  | Processo Acústico. ....                                 | 71 |



# Lista de Tabelas

|     |   |    |
|-----|---|----|
| 2.1 | Notação utilizada.....  | 5  |
| 4.1 | Medições realizadas para obtenção da expressão que relaciona $z$ e $f$ . .... | 34 |
| 4.2 | Medidas de profundidade. ....   | 41 |
| 4.3 | Ciclo de comunicações. ....   | 46 |
| 6.1 | Posições testadas. ....   | 59 |



# Abreviaturas e Símbolos

|     |                                      |
|-----|--------------------------------------|
| OSG | <i>Ocean Systems Group</i>           |
| ASV | <i>Autonomous Surface Vehicles</i>   |
| AUV | <i>Autonomous Underwater Vehicle</i> |
| EKF | <i>Extended Kalman Filter</i>        |
| RLS | <i>Recursive Least Squares</i>       |
| RGB | <i>Red Green Blue</i>                |
| LBL | <i>Long baseline</i>                 |
| SBL | <i>Short Baseline</i>                |
| UDP | <i>User Datagram Protocol</i>        |
| TCP | <i>Transmission Control Protocol</i> |
| IP  | <i>Internet Protocol</i>             |



# Capítulo 1

## Introdução

### 1.1 Motivação

O projeto enquadra-se nas atividades da Unidade de Robótica e Sistemas Inteligentes do INESC TEC como resposta à crescente procura de soluções robóticas para o ambiente marinho e, em particular, na utilização dessas soluções em ambientes cada vez mais complexos.

Para a realização dos vários trabalhos de investigação com que se depara, o OSG, *Ocean Systems Group*, dispõe de soluções inteiramente desenvolvidas nos seus laboratórios e que passam por veículos aquáticos autónomos capazes de levar a cabo um conjunto de missões programadas.

### 1.2 Objetivos

O objetivo desta dissertação é projetar e desenvolver um sistema de seguimento de um alvo submerso conjugando dois tipos de posicionamento, acústico e óptico. O interesse da utilização de ambos prende-se com o facto de o seguimento ser feito em curto e longo alcance.

O posicionamento acústico tem como vantagem o seu longo alcance e, em contrapartida, pouca exatidão. Já o posicionamento óptico tem como principal vantagem a sua elevada precisão, contudo possui curto alcance. Assim, é possível conciliar as suas vantagens de modo a ocultar as suas desvantagens. O objetivo será dotar veículos submarinos da capacidade de acostar a estações ou seguir alvos móveis.

É possível então enumerar os objetivos principais desta dissertação:

- Estudo e implementação de um sistema acústico de localização de alvo;
- Estudo, desenvolvimento e implementação de um sistema óptico de localização de proximidade;
- Estudo, desenvolvimento e implementação de algoritmos de fusão de posicionamento acústico e óptico;
- Teste do sistema final em condições reais.

O OSG tem já uma estrutura computacional bem definida, bem como um modelo de comunicações específico. Desta forma, todo o trabalho desenvolvido deverá ter em consideração uma integração com os sistemas existentes.

## 1.3 Estrutura do Documento

Este documento está dividido em 6 capítulos.

No capítulo 2 é apresentado algum estado da arte no que diz respeito à visão computacional e navegação acústica. São também introduzidos alguns conceitos teóricos que foram necessários para o desenvolvimento deste trabalho.

No capítulo 3 é referida a estrutura do *Ocean Systems Group*, que vai desde os veículos em si até ao Modelo de Comunicações estabelecido.

No capítulo 4 são descritos os algoritmos desenvolvidos e é discutido o seu funcionamento.

No capítulo 5 é apresentada a implementação.

No capítulo 6 são expostos os resultados obtidos. A conclusão e a discussão de trabalho futuro são abordadas no capítulo 7.

Por fim, em anexo estão disponíveis diagramas representativos dos algoritmos.



## Capítulo 2

### Estado da Arte

#### 2.1 Modelo do sistema

A existência de um modelo do sistema é importante, uma vez que permite um conhecimento profundo do processo a ser utilizado, bem como das suas limitações, permitindo uma avaliação mais rigorosa dos desempenhos do sistema. Neste projeto em concreto serão utilizados veículos marinhos. Apesar de não ser um dos objetivos projetar um controlador, é benéfico ter conhecimento deste, visto que é uma parte crucial para o movimento do veículo.

Considera-se um veículo marinho como um corpo rígido, admitindo que este é indeformável e que não se criam forças entre os seus elementos individuais. É possível admitir a existência de um referencial inercial na terra.

Em termos de referenciais é comum usar dois referenciais distintos para descrever o movimento dos veículos e derivar a sua dinâmica. Em primeiro lugar define-se um referencial inercial, desprezando-se para isso as acelerações. Este referencial usa um sistema de coordenadas do tipo NED (*North - East - Down*) significando isto que o eixo das abcissas aponta para Norte, o eixo das ordenadas aponta para Este e o eixo das alturas aponta para baixo. Deve-se ainda definir um referencial móvel com origem coincidente com o centro de gravidade do veículo, com as abcissas orientadas segundo o eixo longitudinal da embarcação, da ré para a popa, sendo as ordenadas segundo o eixo transversal, com direção para estibordo, e a altura, o eixo normal, orientado do topo para o fundo do veículo [1].

O referencial inercial é utilizado para aplicações de localização e o referencial móvel, solidário com o veículo, é útil para derivar a dinâmica do veículo e as suas equações do movimento, já que permite tirar partido de propriedades geométricas do próprio veículo.



Na superfície, a trajetória do ASV (*autonomous surface vehicle*) é feita no plano horizontal. Esta é controlada por duas entradas: a “cabeça” do ASV e a velocidade. Assim, o controlador do sistema de seguimento para o veículo é composto por um controlador de posição e por um controlador de velocidade. Para controlar a “cabeça” do veículo é necessário que este seja informado das coordenadas do objeto a seguir e para o controlador da velocidade é essencial saber a velocidade a que o objeto navega. No caso de o objeto a seguir estar a uma distância muito grande do ASV, a estratégia adotada é a de seguir diretamente em direção ao objeto à velocidade máxima [2].

**Tabela 2.1** – Notação utilizada

| DOF |                                      | Forças<br>e<br>Momentos | Velocidades<br>linear e angular | Posição e<br>Ângulos de<br>Euler |
|-----|--------------------------------------|-------------------------|---------------------------------|----------------------------------|
| 1   | trans. na direção x( <i>surge</i> )  | X                       | n                               | x                                |
| 2   | trans. na direção y( <i>sway</i> )   | Y                       | v                               | y                                |
| 3   | trans. na direção z ( <i>heave</i> ) | Z                       | w                               | z                                |
| 4   | rot. em torno de x ( <i>roll</i> )   | K                       | p                               | $\phi$                           |
| 5   | rot. em torno de y ( <i>pitch</i> )  | M                       | q                               | $\theta$                         |
| 6   | rot. em torno de z ( <i>yaw</i> )    | N                       | r                               | $\psi$                           |

Para descrever completamente o movimento de um corpo é necessário saber determinar a sua posição e a sua orientação relativamente a um referencial. Dessa forma qualquer modelo que se pretenda representar precisamente terá 6 graus de liberdade, 3 relativos à posição e movimentos de translação e os restantes 3 relativos à orientação e rotação do corpo.

Com base na segunda lei de Newton e nos Axiomas de Euler é possível estabelecer um modelo com 6 graus de liberdade para a dinâmica de um corpo rígido, tal como se pode observar na equação 2.1 [3],

$$M\dot{v} + C(v)v + Dv = \tau, \quad (2.1)$$

M representa a matriz de inércia do corpo rígido, C é a matriz das forças de Coriolis e D a Centrípeta, provocadas pela existência de um referencial girante.

$$M = \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{bmatrix}, \quad D = \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & 0 \\ 0 & 0 & d_{33} \end{bmatrix}, \quad (2.2)$$

$$C(v) = \begin{bmatrix} 0 & 0 & -m(x_g r + v) \\ 0 & 0 & mu \\ m(x_g r + v) & -mu & 0 \end{bmatrix} \quad (2.3)$$

onde  $m$  é a massa do ASV e  $I_z$  o momento de inércia em torno do eixo dos  $z$ 's. A diagonal de  $D$  representa o amortecimento,  $v = [u; v; w; p; q; r]^T$  é o vetor das velocidades lineares e angulares e  $\tau$  representa o vetor de todas as forças externas e momentos aplicados ao corpo rígido, forças essas como as forças de massa e inércia, de amortecimento hidrodinâmico e forças criadas pela atuação dos motores do barco [3].

É então possível obter um sistema de equações mais simples, com apenas 3 graus de liberdade, e que sintetiza o movimento de um veículo marinho de superfície,

$$X = m(\dot{u} - vr), \quad (2.4)$$

$$Y = m(\dot{v} - ur), \quad (2.5)$$

$$N = I_z \dot{r}. \quad (2.6)$$

Desprezando os efeitos do vento e considerando que a componente horizontal da corrente de água são constantes,  $u$  e  $v$  podem ser tiradas a partir das componentes da velocidade do veículo em relação à água. Neste caso, as forças externas  $X$  e  $Y$  só dependem da velocidade do veículo em relação à água e sobre a atuação do veículo [2]. Além disso, a evolução da posição do veículo num referencial fixo da terra e com a “cabeça” do veículo em relação ao mesmo referencial são dados por:

$$\dot{x} = u \cos \psi - v \sin \psi + c_x, \quad (2.7)$$

$$\dot{y} = u \sin \psi + v \cos \psi + c_y, \quad (2.8)$$

$$\dot{\psi} = r, \quad (2.9)$$

onde  $c_x$  e  $c_y$  são as componentes  $x$  e  $y$  da corrente da água. Dada a simetria do veículo, assumir-se-á que  $X$ ,  $Y$  e  $N$  são dados por:

$$X = X_{\dot{u}} \dot{u} + X_u u + X_{u|u}|u| + X_{act}, \quad (2.10)$$

$$Y = Y_{\dot{v}} \dot{v} + Y_v v + X_{v|v}|v|, \quad (2.11)$$

$$N = N_{\dot{r}} \dot{r} + N_r r + N_{r|r}|r| + N_{act}. \quad (2.12)$$

## 2.2 Método de localização acústica

O sistema de posicionamento baseado em sinais acústicos é habitualmente utilizado em meios aquáticos. A sua utilização deve-se ao facto de a água apresentar boas propriedades de transmissão de ondas acústicas.

O seu princípio de funcionamento baseia-se na troca de sinais acústicos entre os veículos autónomos e cada uma das boias ou faróis acústicos que constituem a rede. A localização de cada boia/farol é conhecida e através da temporização da troca de mensagens, juntamente com o conhecimento da velocidade da propagação das ondas acústicas na água, é possível restringir a posição do veículo a uma superfície.

Na navegação acústica existem diversos métodos que diferem fundamentalmente na geometria e no número de faróis acústicos utilizados. Destes, o LBL e o SBL são os mais frequentemente utilizados, já que são os que atingem desempenhos mais satisfatórios. No entanto, cada um destes tem aplicações distintas.

O SBL (*Short Baseline*) é um método bastante simples, não só ao nível dos recursos utilizados como também ao nível da implementação. Apesar de requerer apenas o uso de um farol acústico, advindo daí a sua simplicidade de operação, é o mais adequado para embarcações de grandes dimensões, já que necessita que cada embarcação tenha no mínimo três *transceivers* colocados no seu casco.

O método de posicionamento acústico LBL, (*Long Baseline*), apesar de ser um sistema mais complexo, visto que necessita de um maior número de faróis acústicos, é também o mais adequado para embarcações de pequenas dimensões. Este é o sistema mais frequentemente utilizado, pois permite atingir precisões bastante elevadas e independentes da profundidade de operação [4].

### 2.2.1 Método LBL

O LBL envolve o uso de um conjunto de faróis acústicos com uma posição fixa e predefinida, e ainda um transdutor acoplado à embarcação cuja posição se pretende saber. A determinação da posição é feita por meio de interrogações às boias, através de um sinal acústico, numa determinada frequência específica para cada boia. Por sua vez, cada um dos faróis acústicos, sempre que deteta um sinal na gama de frequências para o qual foi previamente programado, envia um sinal de resposta para a embarcação. Temporizando cada um destes eventos acústicos é então possível determinar o tempo decorrido entre a interrogação a cada uma das boias e a receção do respetivo sinal de resposta. Conhecendo a velocidade dos sinais acústicos na água é possível calcular a distância em linha reta a cada uma das boias [4].

Para se obter um posicionamento sem incertezas em todas as três dimensões de interesse, é necessário o uso de três faróis acústicos distintos. No entanto, é aconselhável ainda o uso de um farol acústico extra, de modo a obter redundância nas medidas obtidas e, estatisticamente, um posicionamento com mais qualidade.

## 2.3 Método de localização óptico

A detecção de um alvo submerso pode ser feita com recurso a uma câmara, para com isso captar imagens do objeto. O método óptico, à partida, é o que menos erro introduz na localização.

Num sistema óptico ou visão, a captação de imagem é o primeiro processo, passando pela análise da imagem e tendo como produto final o máximo de informações retiradas dessa mesma imagem.

Quando uma imagem RGB é capturada passa pelos seguintes processos:

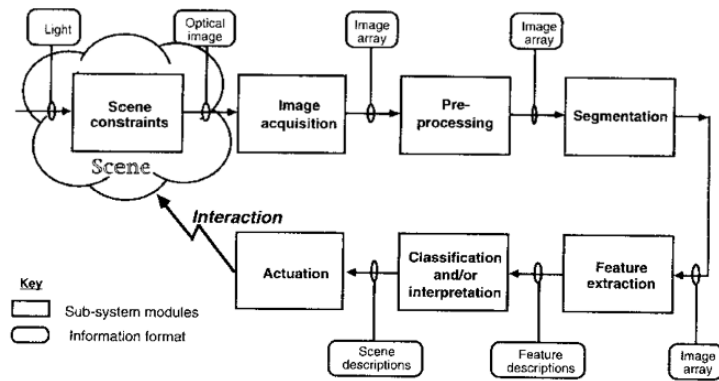
**Pré-processamento** - melhoramento dos dados da imagem através da supressão de degradações indesejadas ou do realce de características importantes.

**Morfologia matemática** - sendo definida como a teoria para análise de estruturas espaciais, visa a analisar a forma dos objetos.

**Segmentação** - é todo o processo de partição da imagem em regiões com atributos semelhantes. Classificação dos pixéis em conjuntos coerentes (regiões/objetos) de acordo com alguma(s) característica(s) comum(ns), intensidade/cor, textura, movimento, etc.

**Análise e representação** - O processo de binarização produz uma grande região de pixéis conectados (objeto de interesse) e uma grande quantidade de pequenas regiões com pixéis conectados (ruído), pelo que é necessário rotular as várias regiões e identificar a região de interesse.

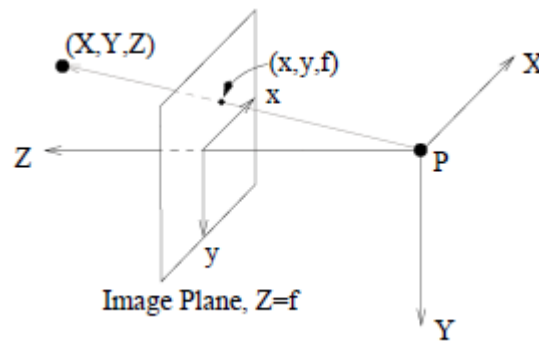
**Reconhecimento e classificação** - O objetivo do reconhecimento de objetos pode ser definido como um problema de rotulagem. Dada uma imagem que contém um ou mais objetos (e fundo) e um conjunto de etiquetas correspondentes a um conjunto de modelos conhecidos, o sistema deve atribuir etiquetas corretas



**Figura 2.1** – Modelo genérico de um sistema baseado em visão [5].

### 2.3.1 Projeção da imagem

A formação da imagem pode ser aproximada por uma câmara *pinhole* simples. A *pinhole* consiste numa maneira de ver uma imagem real, através de uma câmara escura. De um pequeno orifício onde a luz é captada para dentro da câmara e sofrendo um movimento de inversão, a imagem é projetada para a parede oposta ao orifício, ficando invertida. Para produzir uma imagem razoavelmente nítida, a abertura tem que ser um furo pequeno, na ordem de 0,5 mm ou menor [6].



**Figura 2.2** – Formação de imagem [7].

A posição da imagem para um ponto 3D  $(X, Y, Z)$  é dada por uma transformação projetiva,

$$\begin{pmatrix} x \\ y \\ f \end{pmatrix} = \frac{f}{Z} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}. \quad (2.13)$$

A distância entre o plano da imagem e o ponto projetado P é chamado de *focal length* ou distância focal e representa-se por  $f$ .

Considerando três tipos de coordenadas: coordenadas mundo, coordenadas da câmara e coordenadas da imagem.

As coordenadas do mundo,  $\vec{X}_w$  são as coordenadas 3D fixas do mundo, que dizem respeito, por exemplo, ao canto de uma sala.

As coordenadas da câmara,  $\vec{X}_c$  são as coordenadas 3D fixas da câmara. A origem das mesmas é o centro de projeção da câmara ( $\vec{d}_w$  nas coordenadas mundiais). O eixo dos Z's considera-se o eixo óptico da câmara.

Por último, as coordenadas da imagem,  $\vec{p}$  são consideradas um vetor  $\vec{p} = (p_1, p_2, 1)^T$ , onde  $p_1$  e  $p_2$  são as coordenadas do pixel do ponto da imagem. A origem é o canto superior esquerdo da imagem. A primeira coordenada,  $p_1$ , aumenta para a direita e  $p_2$  aumenta para baixo [7].

Depois de considerar estas três coordenadas, pode-se explicar a transformação entre as coordenadas mundiais e as coordenadas da câmara e entre a última e as coordenadas da imagem.

### Matriz de calibração extrínseca

Os parâmetros de calibração extrínseca especificam a transformação das coordenadas mundiais para as coordenadas da câmara, que é uma transformação de coordenadas 3D.

$$\vec{X}_c = M_{ex} [\vec{X}_w^T, 1]^T, \quad (2.14)$$

onde a matriz de calibração extrínseca  $M_{ex}$  é uma matriz 3 x 4,

$$M_{ex} = (R - R\vec{d}_w), \quad (2.15)$$

sendo  $R$  uma matriz rotacional 3 x 3 e  $\vec{d}_w$  é a localização, nas coordenadas mundiais, do centro de projeção da câmara. O mapeamento inverso é simplesmente,

$$\vec{X}_w = R^T \vec{X}_c + \vec{d}_w. \quad (2.16)$$

Agora já se pode aplicar a transformação de perspectiva ao ponto 3D  $\vec{X}_c$  [7],

$$\vec{x}_c = \frac{f}{X_{3,c}} \vec{X}_c = \begin{pmatrix} x_{1,c} \\ x_{2,c} \\ f \end{pmatrix}. \quad (2.17)$$

Nota: Tudo aqui apresentado é mensurável em metros.

### Matriz de calibração intrínseca

A matriz de calibração intrínseca transforma as coordenadas da câmara em coordenadas pixel ou imagem,

$$\vec{p} = \frac{1}{f} M_{in} \vec{x}_c, \quad (2.18)$$

onde  $M_{in}$  é uma matriz 3 x 3.

Para melhor compreensão de  $M_{in}$  tome-se como exemplo uma câmara retangular com tamanho de pixéis  $\frac{1}{s_x}$  por  $\frac{1}{s_y}$  com distância focal  $f$  e o ponto de perfuração da luz  $(O_x, O_y)$ , a sua matriz de calibração intrínseca será,

$$M_{in} = \begin{pmatrix} fs_x & 0 & O_x \\ 0 & fs_y & O_y \\ 0 & 0 & 1 \end{pmatrix}. \quad (2.19)$$

De notar que até agora a distância focal foi referida em metros. Mas os únicos termos a aparecerem na matriz de calibração intrínseca são  $fs_x$  e  $fs_y$  onde  $s_x$  e  $s_y$  estão em pixéis por metro e  $O_x$  e  $O_y$  em pixéis. Assim, em vez de metros, é comum medir  $f$  em largura de pixéis, substituindo  $fs_x$  por  $f$ . Então a matriz fica,

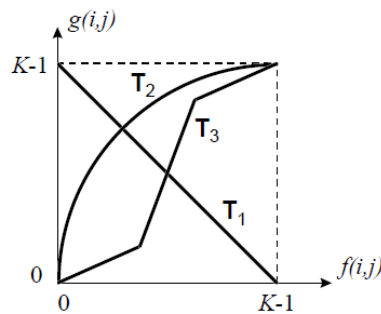
$$M_{in} = \begin{pmatrix} f & 0 & O_x \\ 0 & f\alpha & O_y \\ 0 & 0 & 1 \end{pmatrix}, \quad (2.20)$$

onde  $\alpha = \frac{s_y}{s_x}$  é a proporção de aspeto do pixel (sem unidade)  $0 < \alpha < 1$  se os pixels são retangulares e lisos,  $\alpha = 1$  se quadrados e  $\alpha > 1$  retangulares e altos [7].

### 2.3.2 Pré-processamento

Nas operações pontuais, cada ponto da imagem-resultado  $g(i,j)$  é obtido por: uma transformação  $T$  do ponto de coordenadas homólogas da imagem-operando  $f(i,j)$ , ou por uma operação  $OP$  entre os pontos de coordenadas homólogas das imagens-operando  $f_1(i,j)$  e  $f_2(i,j)$ .

A transformação  $T$  deve ser uma função unívoca com um domínio que é a gama de valores permitidos para a imagem original. Essa transformação pode ser realizada por meio de tabelas de transformação LUT's (*Look-Up Tables*) e interpretada com o auxílio de um diagrama como o da figura 2.3.



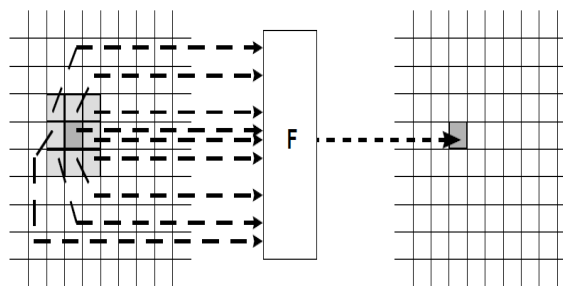
**Figura 2.3** – Diagrama de transformação.

A operação  $OP$  pode ser uma qualquer operação (aritmética, lógica, de comparação, etc) admitida pela natureza dos valores dos pontos das imagens.

Nas operações locais, cada ponto da imagem-resultado é obtido pela aplicação de uma função de transformação  $F$  que usa como argumentos o ponto homólogo da imagem original e um conjunto de pontos vizinhos desse.

Frequentemente a vizinhança considerada consiste numa janela de dimensão  $K \times K$ . Na figura 2.4 abaixo, usa-se uma janela  $3 \times 3$  centrada no ponto com coordenadas idênticas às do ponto-resultado.

A função  $F$  pode ser a combinação linear dos valores dos pontos vistos através da janela (caso em que a operação se designa por convolução), mas também pode ser uma certa função lógica ou não-linear dos mesmos valores [8][10][11][12].



**Figura 2.4** – Processo de transformação de uma operação local.



No caso da convolução, os coeficientes (ou pesos) da combinação linear dos pontos são escolhidos de modo a que a operação local correspondente tenha o efeito pretendido.

Cada ponto-resultado  $g(i,j)$  será obtido por meio de  $g(i,j) = \sum_k w_k \times f_k(i,j)$  em que cada  $f_k(i,j)$  corresponde ao ponto da imagem com posição idêntica na janela, ao coeficiente com o mesmo índice, como na figura 2.5, para o caso 3x3.

|       |       |       |
|-------|-------|-------|
| $w_1$ | $w_2$ | $w_3$ |
| $w_4$ | $w_5$ | $w_6$ |
| $w_7$ | $w_8$ | $w_9$ |

$$g(i,j) = w_1 f(i-1,j-1) + w_2 f(i,j-1) + w_3 f(i+1,j-1) +$$

$$+ w_4 f(i-1,j) + w_5 f(i,j) + w_6 f(i+1,j) +$$

$$+ w_7 f(i-1,j+1) + w_8 f(i,j+1) + w_9 f(i+1,j+1)$$

**Figura 2.5** – Exemplo de uma operação local.

### 2.3.3 Segmentação

A segmentação pode ser considerada como um processo de classificação de pixéis. Se as classes e as suas probabilidades, bem como as propriedades (por exemplo brilho) e as suas probabilidades por classes, forem conhecidas à priori, a segmentação reduz-se a um problema clássico de decisão estatística. Quando as classes não são conhecidas pode recorrer-se a técnicas de agrupamento (*clustering*).

A classificação é muitas vezes baseada no histograma de uma propriedade (em geral, o nível de cinzento). Dado o número de pixéis numa imagem ser usualmente muito elevado pode-se considerar o histograma como uma boa aproximação à densidade de probabilidade da propriedade que ele representa.

#### **Método de *Thresholding* com um único limiar de separação (binarização)**

A escolha do limiar de separação (separação entre as classes “claro” e “escuro” no caso da propriedade ser o brilho) faz-se geralmente no vale do histograma sendo este bimodal [8],

$$B[i,j] \begin{cases} 1 & T_1 < F[i,j] \\ 0 & \text{outro caso} \end{cases}. \quad (2.21)$$

Uma outra forma de segmentação pode ser usada quando se conhecem os objetos ou estruturas presentes na imagem. Trata-se de procurar na imagem uma instância de um modelo pré-determinado, ou seja, estabelecimento de correspondências (*matching*).

Este método tem inúmeras aplicações, nomeadamente:

- Alinhamento de diferentes imagens da mesma cena;
- Detecção de “protótipos” (*template matching*);
- Detecção de características (*feature matching*);
- Correspondência em sequências de imagens.

### **Método *Matching***

Dispõe-se de um padrão  $f(i,j)$  cuja ocorrência na imagem  $g(i,j)$  se procura detetar. Para o efeito desloca-se o padrão para todas as posições possíveis na imagem e mede-se, em cada posição, o grau de semelhança entre o padrão e a imagem, na área em que se sobrepõem (A) [8][9].

Medidas de dissemelhança (*mismatch*) comuns são:

$$\max_A |f - g| \quad \sum \sum_A |f - g| \quad \sum \sum_A (f - g)^2 \quad (2.22)$$

### **Sobel**

O operador de Sobel é um operador de diferenciação discreta. Calcula uma aproximação do gradiente da função de intensidade de uma imagem. O operador de Sobel combina a suavização e a diferenciação Gaussiana [13].

Assumindo a aplicação do operador a uma imagem A, primeiro calcula-se duas derivadas:

- Variações horizontais: são calculadas a partir da convolução de A com um filtro  $G_x$ .

Um exemplo de um filtro de tamanho 3:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A$$

- Variações verticais: são calculadas a partir da convolução de A com  $G_y$ . Um exemplo de um filtro de tamanho 3:

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * A$$

Em cada ponto da imagem calcula-se uma aproximação do gradiente através da combinação dos resultados obtidos com a aplicação de cada filtro,

$$G = \sqrt{G_x^2 + G_y^2}. \quad (2.23)$$

Algumas vezes é usada a seguinte equação em vez da anterior:

$$G = |G_x| + |G_y|. \quad (2.24)$$

### ***Canny Edge Detector***

Canny baseia-se na satisfação de 3 requisitos básicos:

- A taxa de erro do método deve ser muito baixa;
- Os pontos de orlas devem estar bem localizados;
- Só deve ser detetado um ponto por orla.

Segundo Canny, o melhor detetor de orlas do tipo degrau é a primeira derivada de um Gaussiano. A generalização deste conceito para duas dimensões envolve o reconhecimento de que esta abordagem 1D deve ser aplicada na direção normal à orla [14].

As etapas do método são as seguintes:

- Suavização da imagem com um filtro Gaussiano, como exemplo de um filtro gaussiano de tamanho 5,

$$K = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}.$$

- Realce de orlas usando um filtro de gradiente. Para isto é seguido o procedimento análogo a Sobel.
- Aplicar um par de máscaras de convolução nas direções x e y,

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix},$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}.$$

- Encontrar a amplitude e direção do gradiente,

$$G = \sqrt{G_x^2 + G_y^2}, \quad (2.25)$$

$$\theta = \arctan\left(\frac{G_y}{G_x}\right), \quad (2.26)$$

a direção anda em torno dos quatro ângulos possíveis: 0, 45, 90 ou 135 graus.

- Aplicação da supressão de não-máximos (na direção do gradiente) à imagem de amplitude do gradiente. Com isto são removidos os pixéis que não são considerados parte de uma orla. Assim, apenas as linhas finas (orlas candidatas) permanecerão.
- Usar *double-thresholding*, ou seja, dois limiares e análise de conectividade para detetar e ligar as orlas.

Se o gradiente do pixel é maior do que o limiar mais elevado, o pixel é aceite como pertencente a uma orla.

Se o gradiente do pixel é menor do que o limiar mais baixo, o pixel é rejeitado.

Se o gradiente do pixel está entre os dois limiares, então será aceite apenas se estiver ligado a um pixel que está acima do limite superior.

Canny recomenda uma relação cima-baixo entre 2:1 e 3:1 [14].

### 2.3.4 Análise e representação

Após segmentação é comum produzir-se uma imagem binária em que diversos objetos disjuntos têm pixéis de valor 1, e se situam sobre um fundo de valor 0.

Para efeitos de análise, é importante discriminar entre os diversos objetos, atribuindo a todos os pixéis de um objeto uma etiqueta exclusiva. Para se conseguir essa etiquetagem (*component labeling*) podem ser usados dois algoritmos básicos.

#### Algoritmo 1

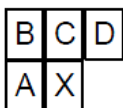
- Efetuar um varrimento normal da imagem até ser encontrado um pixel de valor 1.
- Atribuir uma nova etiqueta ao pixel encontrado.
- Partindo desse pixel, propagar a etiqueta para os pixéis vizinhos de valor 1, em todas as direções, até esgotar o componente.
- Prosseguir o varrimento, a partir do ponto onde foi interrompido, e proceder como anteriormente para novos componentes.
- Terminar quando se completar o varrimento.

Este algoritmo tem o inconveniente de exigir o acesso aleatório aos pixéis da imagem, podendo ser de execução bastante lenta.

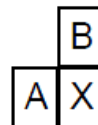
#### Algoritmo 2

- Efetuar um varrimento normal da imagem.
- Para cada ponto 1 encontrado (X), analisar os vizinhos imediatos já visitados, como se expõe a seguir.

Conectividade-8



Conectividade-4



- Se todos esses vizinhos forem 0, atribuir a X uma etiqueta nova.
- Caso contrário, lendo as etiquetas de A, B, C e D (ou apenas de A e B, para conectividade-4) por essa ordem, atribuir a X a primeira etiqueta diferente de 0.

- Se entre os pontos A, B, C e D houver mais do que uma etiqueta, assinalar a equivalência dessas etiquetas numa lista.
- Terminado o varrimento, processar a lista de equivalências, definindo uma etiqueta única para cada grupo de etiquetas equivalentes.
- Efetuar um segundo varrimento da imagem, atribuindo aos pixéis as novas etiquetas.

### 2.3.5 Reconhecimento e classificação

Completada a segmentação de uma imagem, a etiquetagem e a representação dos segmentos, é necessário descrever, reconhecer, identificar ou classificar essas entidades. Para esse efeito, a extração e a medida de características dos segmentos constitui uma fase indispensável.

De entre as inúmeras características que se podem extrair, enunciam-se de seguida algumas das mais comuns.

**Área** – geralmente tomada como o número de pixéis que constituem um segmento. Se tiver sido feita a calibração do sistema de aquisição de imagem, o número de pixéis pode ser convertido para medidas de área no referencial-objeto (por exemplo, mm<sup>2</sup>).

**Perímetro** – é o comprimento do(s) contorno(s) de um segmento. Pode ser medido pelo número de pixéis que constituem o contorno, obtido quer com conectividade-4 quer com conectividade-8. Muitas vezes para se conseguir uma melhor aproximação ao perímetro euclidiano, toma-se o comprimento 1 para a distância entre pixéis contíguos horizontal ou verticalmente, e o comprimento  $\sqrt{2}$  para a distância diagonal. No caso de existir informação de calibração, o perímetro pode também ser convertido para unidades do referencial-objeto.

**“Redondez” (roundness, compactness)** – medida baseada na relação entre área e perímetro de um objeto. Trata-se de uma medida adimensional, calculada como,

$$\frac{\text{Perímetro}^2}{4\pi \times \text{Área}}. \quad (2.27)$$

O valor desta medida é mínimo (com o valor 1) para objetos perfeitamente circulares (em geometria contínua euclidiana), tomando valores muito mais elevados para objetos alongados ou muito recortados.

**Número de Euler** – parâmetro topológico que para uma imagem binária tem o valor

$$E = C - H, \quad (2.28)$$

em que C é o número de segmentos conexos e H é o número de furos nesses segmentos.

**Retângulo envolvente** – é o retângulo de lados horizontais e verticais que circunscreve o segmento. Os lados e a área desse retângulo podem ser usados como características, embora apresentem valores muito sensíveis à rotação dos objetos.

**Retângulo “equivalente”** – é o retângulo de lados *length* e *breadth* que tem a mesma área e perímetro que o objeto a que respeita. Note-se que, no caso de objetos compactos (“redondez” próxima de 1), pode não existir o retângulo “equivalente”.

**Raios máximo e mínimo** – são as distâncias do centróide do objeto aos pontos mais afastados e mais próximos do contorno exterior do objeto, respetivamente. A razão entre estes dois parâmetros (raio máximo/raio mínimo) define uma característica descritiva também importante, designada por excentricidade.

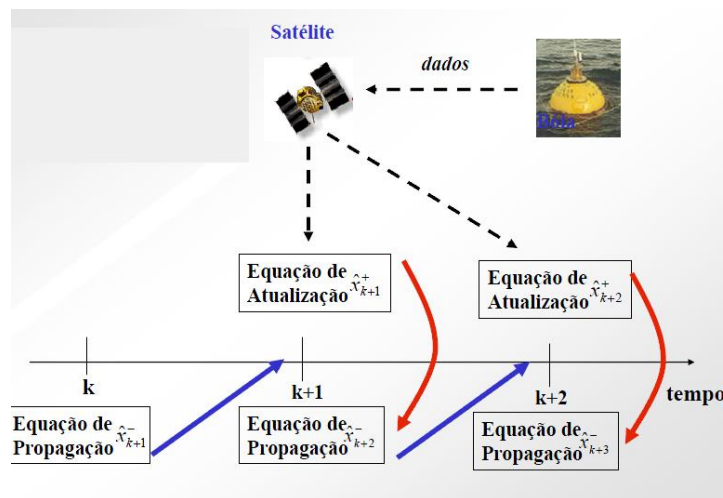
**Orientação** – a orientação é o ângulo formado entre a horizontal e a direção do eixo do objeto com momento de inércia mínimo. A minimização do momento de inércia

$$I(\theta) = \sum_{(i,j) \in K} D^2(i,j) = \sum_{(i,j) \in K} [(j - y) \cos(\theta) - (i - x) \sin(\theta)]^2, \quad (2.29)$$

permite obter o valor da orientação a partir de momentos centrais de 2ª ordem:

$$\theta = \frac{1}{2} \arctg \frac{2\mu_{11}}{\mu_{20} - \mu_{02}}. \quad (2.30)$$

## 2.4 Filtro de Kalman



**Figura 2.6** – Sistema genérico de aplicação do Filtro de Kalman [17].

O estado de um sistema é um conjunto de variáveis que contém a sua memória. Uma representação no espaço de estados é um modelo matemático de um sistema dinâmico que relaciona as suas variáveis de estado com as suas entradas e saídas através de uma equação diferencial (no caso de sistemas contínuos) ou às diferenças (no caso de sistemas discretos) de primeira ordem. No caso de sistemas discretos, lineares e invariantes no tempo, estas representações são do tipo:  $x(t+1) = Ax(t) + Bu(t)$  e  $y(t) = Cx(t) + Du(t)$  onde  $y(t) \in \mathbb{R}^l$  é a saída,  $u(t) \in \mathbb{R}^m$  a entrada e  $x(t) \in \mathbb{R}^n$  vetor das variáveis de estado. As matrizes  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{l \times n}$ , e  $D \in \mathbb{R}^{l \times m}$  são os parâmetros do sistema [18].

O conhecimento das variáveis de estado  $x(t)$  é importante para resolver muitos problemas, tais como a estabilização de sistemas. No entanto, nem sempre é possível medir as variáveis de estado através dos sensores, porque ou esses sensores não existem ou porque existe um intervalo de tempo entre aquisição e processamento elevado.

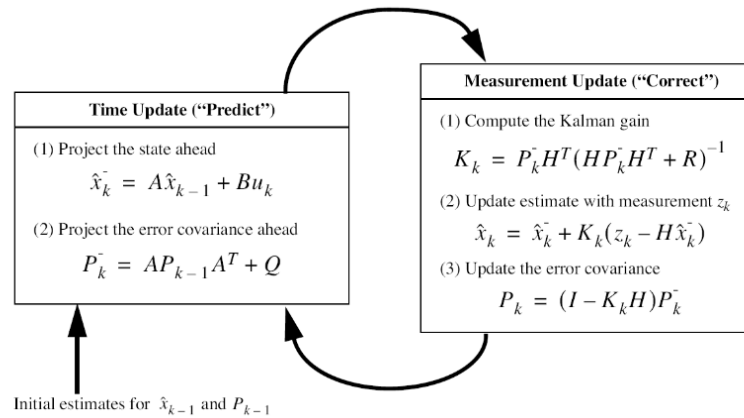
Para resolver este problema, procura-se estimar o valor do estado através da observação dos sinais de entrada e de saída do sistema, utilizando um observador (ou estimador) de estado [18].

Uma das razões para a grande popularidade dos Filtros de Kalman nas áreas da robótica prende-se com o facto de o Filtro de Kalman ser um estimador ótimo de estado de sistemas dinâmicos,

$$X_K = AX_{K-1} + Bu_K + w_{k-1}, X \in \mathfrak{R}^n, \quad (2.31)$$

$$Z_K = HX_K + v_K, Z \in \mathfrak{R}^m. \quad (2.32)$$

O filtro de Kalman funciona com duas etapas distintas, tendo para isso dois conjuntos de equações distintas: previsão do estado e correção do estado.



**Figura 2.7** – Ciclo de processamento do Filtro de Kalman.



Numa primeira fase, como visto na figura 2.7, o filtro de Kalman estima à priori o estado por intermédio da dinâmica conhecida do sistema e pela sua covariância do erro, tendo por base os estados anteriores. Nestas equações, a matriz  $Q$  representa a covariância do ruído da evolução do sistema. Numa segunda fase, e sempre que se dispõe de uma nova medida de estado, estas estimativas são atualizadas e corrigidas.

## 2.5 Filtro de Kalman Estendido

O filtro de Kalman estendido (EKF) calcula um valor próximo do real, uma vez que é usado em sistemas não lineares.

A principal ideia na aplicação do EKF é a linearização. Suponha-se que se tem uma função  $f()$  não linear. A curva Gaussiana projetada através desta função é tipicamente não Gaussiana. Isto acontece porque a não linearidade de  $f()$  distorce a forma Gaussiana para o próximo estado. A linearização aproxima  $f()$  por uma função linear que é tangente à média da curva Gaussiana. Ao projetar a curva Gaussiana através desta aproximação linear, a certeza posterior é Gaussiana. De facto, uma vez que  $f()$  é linearizada, a mecânica de propagação da certeza é equivalente à do filtro de Kalman. O EKF utiliza um método de Expansão de Taylor (1ª ordem). A expansão de Taylor constrói uma aproximação linear da função  $f()$  a partir do valor e declive. Este declive é dado pela derivação parcial da função  $f()$  [20],

$$f'(x_k, u_{k+1}) = \frac{\partial f(x_k, u_{k+1})}{\partial x_k} \quad (2.33)$$

As linearizações de  $f()$  e  $h()$  em relação às variáveis de estado e ao ruído do processo são usadas pelo algoritmo EKF. Estas são denominadas de matrizes Jacobianas e podem ser vistas como mostra a figura 2.8.

$$F = Df(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

$$H = Dh(x) = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \dots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \dots & \frac{\partial h_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_n}{\partial x_1} & \frac{\partial h_n}{\partial x_2} & \dots & \frac{\partial h_n}{\partial x_n} \end{bmatrix}$$

$$W = Df(w) = \begin{bmatrix} \frac{\partial f_1}{\partial w_1} & \frac{\partial f_1}{\partial w_2} & \dots & \frac{\partial f_1}{\partial w_n} \\ \frac{\partial f_2}{\partial w_1} & \frac{\partial f_2}{\partial w_2} & \dots & \frac{\partial f_2}{\partial w_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial w_1} & \frac{\partial f_n}{\partial w_2} & \dots & \frac{\partial f_n}{\partial w_n} \end{bmatrix}$$

**Figura 2.8** – Matrizes Jacobianas.

Modelo discreto não linear,

$$x_k = f(x_{k-1}, u_k, w_{k-1}), \quad (2.34)$$

$$z_k = h(x_k, v_k), \quad (2.35)$$

Substituindo o ruído pela sua média,

$$\tilde{x}_k = f(\tilde{x}_{k-1}, u_k, 0). \quad (2.36)$$

Aproximação linear,

$$x_k \approx \tilde{x}_k + A(x_{k-1} - \hat{x}_{k+1}) + Ww_{k+1}, \quad (2.37)$$

$$z_k \approx \tilde{z}_k + H(x_k - \tilde{x}_k) + Vv_k, \quad (2.38)$$

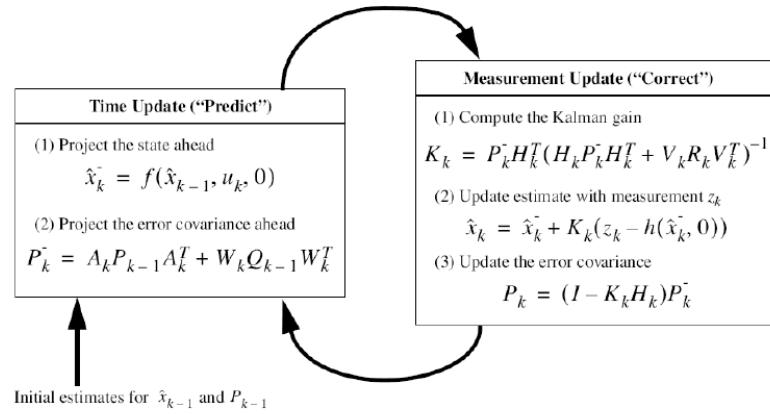
com,

$$A[i, j] = \frac{\partial f[i]}{\partial x[j]}(\hat{x}_{k-1}, u_k, 0), \quad (2.39)$$

$$H[i, j] = \frac{\partial h[i]}{\partial x[j]}(\tilde{x}_k, 0), \quad (2.40)$$

$$W[i, j] = \frac{\partial f[i]}{\partial w[j]}(\hat{x}_{k-1}, u_k, 0), \quad (2.41)$$

$$V[i, j] = \frac{\partial h[i]}{\partial v[j]}(\tilde{x}_k, 0). \quad (2.42)$$



**Figura 2.9** – Ciclo de processamento do Filtro de Kalman Estendido.

Na figura 2.9 é possível ver o ciclo de processamento do Filtro de Kalman Estendido.



## Capítulo 3

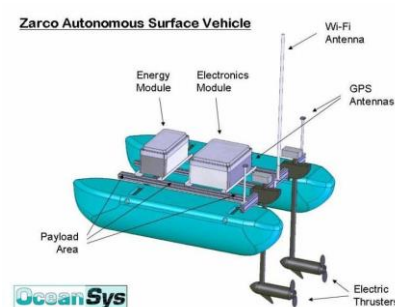
# Plataformas

### 3.1 ASV Zarco

O veículo autónomo de superfície ASV Zarco é uma embarcação de pequena dimensão, ao estilo *catamaran*, concebido para operações a desempenhar em águas com pouca agitação. Este veículo, alimentado por um conjunto de baterias que lhe confere uma autonomia que pode chegar até 3 horas, contém um conjunto de dois motores colocados na sua popa, atingindo velocidades na ordem dos 2m/s.

O ASV pode ser autonomamente controlado, seguindo rotas previamente predefinidas ou, em alternativa, pode ainda ser controlado remotamente, através de uma ligação WiFi com a estação terra.

Para o controlo autónomo, este veículo está dotado de sensores de navegação necessários, onde se incluem um recetor GPS de elevada precisão. Este veículo poderá ainda operar como um farol acústico móvel, já que dispõe de recetores e emissores acústicos subaquáticos [21].



**Figura 3.1** - ASV Zarco [21].

## 3.2 AUV MARES

O AUV MARES é um veículo autónomo subaquático, implementado com o objetivo de recolha de dados oceanográficos em diferentes ambientes, nomeadamente estuários de rios e lagos, tendo também já sido testado o seu funcionamento com sucesso em alto mar.

As missões que se pretende que o AUV desempenhe são de natureza bastante diversa. Deste modo, o AUV foi concebido com uma estrutura altamente modular, permitindo alterar a configuração dos diversos sensores facilmente. Para além dos sensores e instrumentação necessária à navegação autónoma, o AUV pode ainda transportar uma variada gama de outros sensores oceanográficos, que podem ser incluídos ou retirados com facilidade, consoante a missão.

As dimensões deste veículo são também um dos seus pontos fortes. O seu comprimento não ultrapassa os 1,5m e o seu peso é de cerca de 32kg, podendo ser facilmente transportável.

O MARES é atuado por um conjunto de quatro pequenos motores elétricos (figura 3.2), colocados estrategicamente na sua estrutura de tal forma que o movimento no plano horizontal é praticamente independente do movimento no plano vertical. Estes motores são alimentados por um conjunto de baterias que fornecem uma autonomia de até 10 horas ou 40km [21].

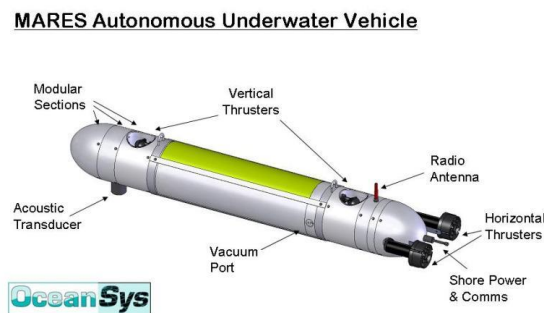


Figura 3.2 – AUV MARES [21].

## 3.3 Boias

As boias, também chamadas de faróis acústicos, são fundamentais na operação dos veículos autónomos subaquáticos, já que por vezes são a única ferramenta de localização que os veículos dispõem.

Cada boia possui, para além de um conjunto de equipamento eletrónico, um transdutor acústico. Este recetor permite que as boias detetem todos os sinais acústicos numa gama de frequências alargada e apropriada para a comunicação subaquática. As frequências estão, geralmente, na gama dos 20KHz até aos 30KHz. No entanto, cada uma das boias encontra-se equipada com um conjunto de filtros programáveis, os quais permitem que cada uma delas distinga uma gama de frequências diferentes, frequências às quais devem responder [21].

Numa missão, o AUV interroga uma das boias com um sinal acústico numa determinada frequência. Apesar de esse sinal ser recebido por todas as boias da rede, no caso de estar à escuta em todas as frequências, apenas uma irá responder com um sinal acústico apropriado. Desta forma, é possível que os veículos interroguem cada uma das boias individualmente, extrapolando daí a sua posição.

Cada evento acústico que ocorra em cada boia é automaticamente enviado para o repositório, juntamente com a identificação da boia que o detetou e ainda com informação relativa ao instante em que aconteceu. Na figura 3.3 é possível ver o aspeto de uma boia acústica.



**Figura 3.3** – Boia acústica [21].

### 3.4 Câmara subaquática

A necessidade de aquisição de imagens subaquáticas para a realização deste projeto levou à escolha e preparação de uma câmara subaquática. A câmara utilizada é uma simples *webcam* da Logitech, embebida num invólucro de plástico de modo a proteger a câmara do ambiente submerso.

A sua escolha deveu-se à facilidade de utilização e baixo custo. A lente da câmara tem uma abertura, aproximadamente, de 40 graus. A figura 3.4 apresenta o seu aspeto.



**Figura 3.4** – Câmera subaquática.

### 3.5 Modelo de Comunicação

De acordo com o Modelo de Comunicação *OceanSys* é estabelecido o conceito de entidade, o qual se refere a qualquer veículo ou sistema de operação presente numa missão. Deste modo, cada um dos veículos, ASV ou AUV, em missão é uma entidade distinta, bem como quaisquer boias que possam integrar a rede acústica. Cada entidade é identificada individualmente pelo seu endereço e poderá ser constituída por um ou mais subsistemas, possuindo cada um o seu endereço individual. Desta forma, é possível uma comunicação entre cada subsistema e entre cada entidade. Exemplificando, o AUV em missão constitui uma entidade e tem diferentes subsistemas, como o sistema de navegação e controlo, sistema de energia, etc. Assim, é possível a qualquer subsistema exterior ao AUV comunicar diretamente com um subsistema em particular.

O Modelo de Comunicação *OceanSys* estabelece também um modo de comunicação, utilizando o protocolo IP como base para a troca de mensagens. Concretamente, esta rede IP é baseada em *routers wireless*, permitindo a comunicação sem fios e à distância entre boias, ASV e estação base.

As mensagens trocadas entre entidades e subsistemas encontram-se encapsuladas em mensagens IP e são compostas por 4 campos: 3 de comprimento fixo, que constituem o cabeçalho da mensagem, e um campo de comprimento variável, que consiste na própria mensagem a transmitir. Para além do campo *MsgType*, que define qual o tipo de mensagem, é ainda constituído por campos de ID que identificam a entidade e subsistema de origem ou destino, dependendo do tipo de mensagem.



## 3.6 Repositório

Como suporte às missões é utilizado um repositório de informações, para onde são canalizados todos os dados considerados relevantes. Este repositório permite, de forma acessível, centralizar todas as informações e comunicações do sistema, possibilitando ainda o seu armazenamento. Este último aspeto é muito importante para detetar e resolver problemas que possam eventualmente surgir no decorrer de cada missão.

O acesso aos dados recebidos pelo repositório é feito do tipo subscrição, ou seja, uma entidade regista-se no repositório, enviando um pedido com o que deseja receber e, a partir desse momento, passa a receber as informações pretendidas. No caso deste trabalho, os algoritmos desenvolvidos necessitam de algumas informações provenientes de outros subsistemas, nomeadamente das boias, dados esses que serão enviados pelo repositório.



## Capítulo 4

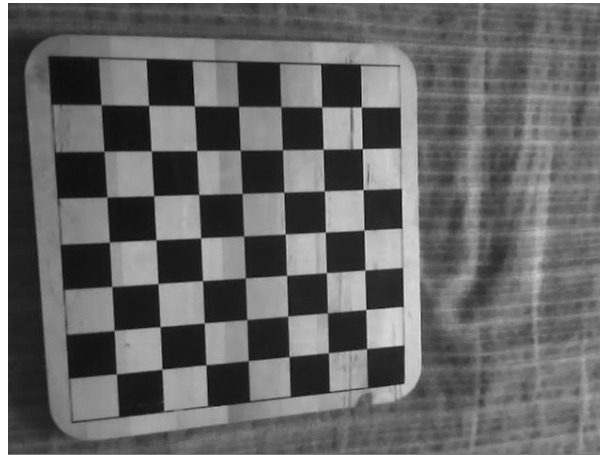
# Trabalho desenvolvido

### 4.1 Calibração

A calibração é um processo essencial quando se trata de visão computacional e existe a necessidade de medir distâncias, obter posições, etc.

A obtenção dos parâmetros intrínsecos necessários para a conversão entre referenciais, como referido em 2.3.1, teve como auxílio a *Camera Calibration toolbox*. O *software* utilizado para este processo foi o matlab, sendo que também era possível a utilização das bibliotecas OpenCV para o mesmo fim. Foi ainda necessário usar a câmara subaquática e um tabuleiro de xadrez.

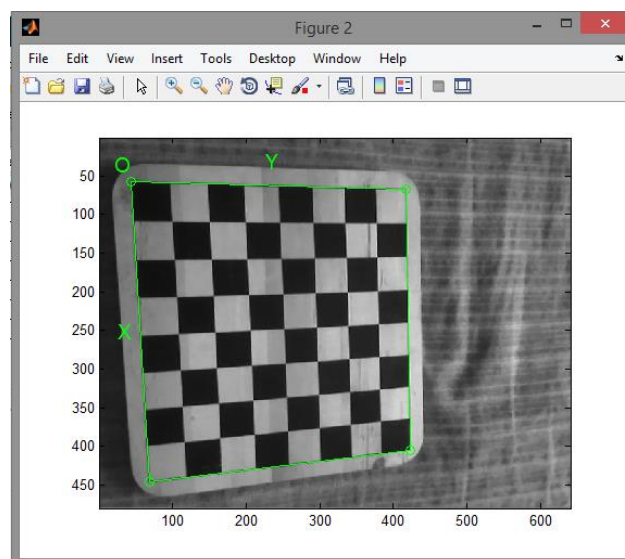
Inicialmente adquiriu-se imagens do tabuleiro em várias posições, utilizando a câmara. Posteriormente converteu-se cada imagem para a escala de cinzentos. Captou-se 12 posições diferentes. Na figura 4.1 está presente um exemplo já convertido para a escala de cinzentos.



**Figura 4.1** – Imagem captada para calibração.

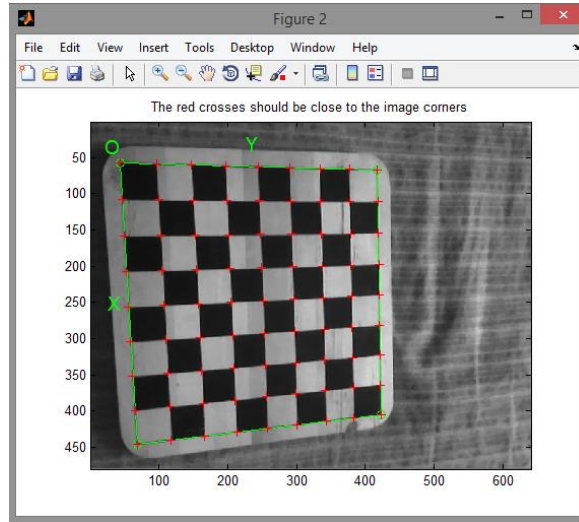
Não existe um número mínimo de imagens necessárias, mas quanto maior for, maior será a aproximação aos valores reais.

Seguidamente carregou-se cada imagem para a toolbox e iniciou-se o processo de extração de cantos. A seleção de cada canto do tabuleiro foi feita manualmente. A figura 4.2 apresenta um exemplo dessa extração de cantos.



**Figura 4.2** – Imagem com a seleção dos cantos.

Posteriormente a *toolbox* encarrega-se de detetar cada quadrado do tabuleiro e de contar o número de quadrados presentes na horizontal e vertical. A figura 4.3 mostra a seleção de cada quadrado.



**Figura 4.3** – Imagem com os quadrados selecionados.

Por fim com auxílio da *toolbox* foram calculados os parâmetros intrínsecos, inclusive a matriz  $M_{in}$ , necessária para a conversão entre os referenciais imagem e câmara, referida em 2.3.1,

$$M_{in} = \begin{bmatrix} 656.004 & 0 & 344.4945 \\ 0 & 656.8646 & 249.9937 \\ 0 & 0 & 1 \end{bmatrix}. \quad (4.1)$$

A conversão entre o referencial da câmara e o referencial mundo pode ser feita através dos parâmetros da calibração extrínseca como explicados em 2.3.1. Devido à sua complexidade seguiu-se outra abordagem. Sabe-se que  $x_c$  e  $y_c$  do referencial da câmara relaciona-se com  $x_m$  e  $y_m$  relativas ao referencial mundo através de um fator multiplicativo,

$$\begin{bmatrix} x_m \\ y_m \end{bmatrix} = \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix} \cdot \begin{bmatrix} x_c \\ y_c \end{bmatrix}. \quad (4.2)$$

O fator  $f$  varia com a distância  $z$ , sendo esta a distância entre a câmara e o alvo. Tendo isto em consideração foi possível calcular  $f$  com base na profundidade do alvo. Com o auxílio do tabuleiro de xadrez foi possível realizar quatro medições com o mesmo a diferentes profundidades. Foi assim possível obter  $f_x$  e  $f_y$  para cada profundidade. A tabela 4.1 contém os resultados obtidos nas medições.

**Tabela 4.1** - Medições realizadas para obtenção da expressão que relaciona  $z$  e  $f$ .

| Medições |        |        |        |        |
|----------|--------|--------|--------|--------|
| Z (m)    | 0.78   | 1.50   | 1.70   | 2.10   |
| $f_x$    | 0.7459 | 1.2787 | 1.3876 | 1.7205 |
| $f_y$    | 0.6509 | 1.3781 | 1.6397 | 1.9827 |

Com isto e através da aproximação por uma reta do método de mínimos quadrados, chegou-se à expressão que relaciona  $z$  e  $f$ ,

$$f_x = 0.7309 \cdot z + 0.1722, \quad (4.3)$$

$$f_y = 1.0208 \cdot z - 0.1388. \quad (4.4)$$

## 4.2 Alvo

A escolha e preparação do alvo é um dos pontos importantes quando se está a lidar com a visão computacional. Este passo facilita bastante os processos de deteção, aquisição e análise de imagens. Sempre que se tem a liberdade de definir a partir do zero um alvo ou marcador é importante que se faça um estudo prévio dos ambientes em que este se vai encontrar, em que condições e a que eventuais ruídos poderá estar sujeito. Isto porque em situações em que o alvo ou marcador não possa ser manipulado ou criado pelo mesmo sujeito que fará a codificação da manipulação de imagens poderá tornar esta codificação mais complicada e pesada computacionalmente.

No caso deste projeto, o alvo foi criado propositadamente para este fim, uma vez que se trata de um marcador que poderá ser adaptado e inserido no objeto em eventuais missões com o objetivo de o seguir. Neste caso seria inserido num veículo subaquático, mais propriamente o AUV MARES.

### 4.2.1 Escolha de propriedades

A necessidade de criar um alvo capaz de sobressair no meio subaquático levou à escolha da cor branca para o mesmo. A cor branca tem como característica uma boa reflexão da luz em

ambientes com pouca luminosidade, como o ambiente submerso. A escolha da forma do alvo teve como critérios a simplicidade e facilidade de detecção e as informações necessárias a retirar deste. Para além da posição definiu-se que a orientação também seria uma informação interessante. Tendo isto em atenção considerou-se a junção de duas formas geométricas, o quadrado ou retângulo com o triângulo. Outro critério fundamental e talvez o mais importante é que o alvo fosse resistente à água, pelo que se decidiu plastificar de modo a obter esta característica.

As dimensões estabelecidas para cada forma geométrica foram de 0.14x0.16 m para o retângulo e 0.07 m de altura para o triângulo isósceles. As dimensões tiveram como critérios a profundidade máxima a que o alvo poderia estar sujeito e a área detetada pela câmara em pixéis. Definiu-se 2 m como profundidade máxima de detecção, pois a partir da mesma a visibilidade torna-se nula ou quase nula. Considerou-se como valor majorante da área de ruído a 2 m de profundidade, 200 pixéis<sup>2</sup>. Tendo este em consideração projetou-se como valor mínimo de área para o alvo, a 2 m de profundidade, de 100 vezes superior à área do ruído. A partir disto e depois da conversão dos valores em pixéis para metros estabeleceu-se as dimensões referidas.

A conversão da área em pixéis de uma profundidade para outra e posteriormente para metros foi realizada com base em testes e aproximações. Primeiramente criou-se um alvo com medidas aleatórias, calculando-se assim a área deste. Com a câmara a 0.2 m obteve-se as medidas em pixéis, incluindo a área. De seguida fez-se novamente uma medição, com o alvo a 2 m. Obteve-se a área deste em pixéis e estabeleceu-se uma relação entre a profundidade e a área. Assim pôde-se obter a percentagem de redução em relação à profundidade. O valor obtido foi de 89.8%. Sabendo que a 2 m de profundidade a área necessária é de 20000 pixéis<sup>2</sup>, calculou-se por fim o valor de área a 0.2 m.

Com base nestes critérios criou-se o primeiro alvo, como se pode observar na figura 4.4.



**Figura 4.4** – Alvo inicial.

#### 4.2.2 Testes

Posteriormente à criação do alvo referido iniciou-se testes de aquisição do alvo submerso e extração de informações com o objetivo de verificar a adequação deste ao projeto. Verificou-se que apesar de se detetar razoavelmente bem ocorreram situações em que o ruído prejudicou a forma do alvo, deformando-o. Isto é, não apresentando a forma retangular na base e triangular no cimo, como expectável. O ruído normalmente verificado consistia em partículas presentes na água que refletiam a luz como a cor branca e, devido ao processo de segmentação da imagem, estas eram detetadas e prejudicavam o processo. Como tentativa de corrigir tal problema usou-se técnicas de manipulação de imagem, tais como erosão, abertura e fecho para colmatar o ruído existente. Estas operações não foram de todo eficazes, pois além de aumentarem o processamento computacional, também em certas situações reduziam o tamanho do alvo ou deformavam-no ainda mais.

#### 4.2.3 Solução final

A solução encontrada passou novamente pela manipulação do alvo, ou seja, alterando-o de modo a ser o mais imune possível ao ruído. Redesenhou-se de novo o alvo, mantendo as formas e cor, acrescentando apenas contornos pretos em torno deste. Os contornos pretos fazem com que exista uma separação entre o alvo e o meio, criando uma espécie de barreira contra o ruído. Desta forma, é possível aplicar filtros de redução de ruído sem prejudicar a forma do alvo.



**Figura 4.5** – Alvo final.



Na figura 4.5 é possível observar os contornos adicionados ao alvo. É possível verificar também que existe uma linha preta entre a forma retangular do alvo e a triangular. Isso deveu-se à necessidade de adquirir separadamente cada forma, com o intuito de facilitar a extração da orientação, tal como será explicado na próxima secção.

## 4.3 Algoritmos desenvolvidos

Esta secção encontra-se dividida em três partes distintas e aborda os diferentes algoritmos implementados. É importante referir que, a partir deste momento e até ao final da dissertação, irão ser referidas duas palavras para a descrição da parte ótica do trabalho, cujo significado atribuído é o mesmo para ambas, nomeadamente ótica e visão. As palavras sistema e subsistema também são mencionadas. Quando se descreve cada parte individualmente é utilizada a palavra sistema e aquando da descrição do sistema completo é referido subsistema.

Na primeira secção é apresentado o sistema óptico que se divide em duas partes, deteção e análise, sendo a última fundamental para a extração de informações relativas à posição do alvo. São ainda discutidas as várias particularidades inerentes a estes processos, bem como soluções que os permitam otimizar.

Numa segunda subsecção é descrito o processo de aquisição de informação por parte do sistema acústico.

Na última subsecção são apresentadas as soluções para a fusão dos dados provenientes dos sistemas acústico e óptico e para o problema da estimação da posição do alvo, fundamental para que os resultados no seu global sejam satisfatórios.

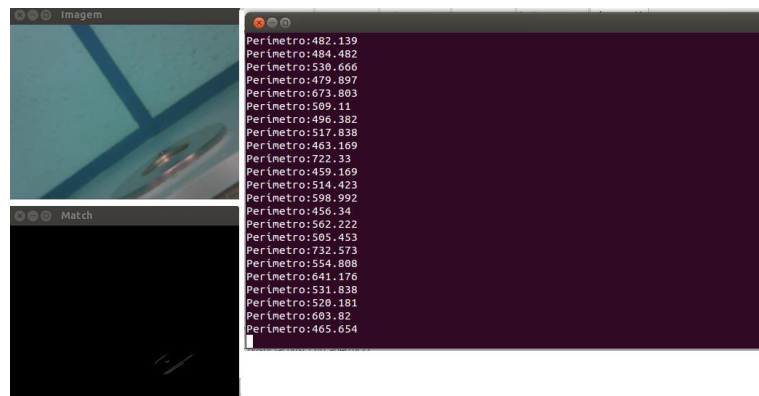
É importante salientar que o material utilizado não permitiu obter com exatidão a profundidade do objeto, uma vez que para tal seria necessário usar três boias acústicas e duas câmaras, respetivamente para obter a profundidade via acústica e via ótica. Contudo, foi possível estimar a profundidade através da câmara disponível.

### 4.3.1 Visão computacional

Tal como foi supramencionado, o algoritmo desenvolvido foi dividido em duas partes. Esta divisão foi necessária, visto que o objetivo desta dissertação é o sistema acústico-óptico, isto é, também utiliza informações recolhidas pelo sistema acústico e, como tal, nem sempre o alvo estará no campo de visão da câmara subaquática. Com isto foi necessário a implementação de um algoritmo que detetasse a presença do alvo. Em caso afirmativo, a segunda parte do algoritmo, a análise, é essencial para a extração de informações relativas à posição  $x$  e  $y$  e orientação do alvo.

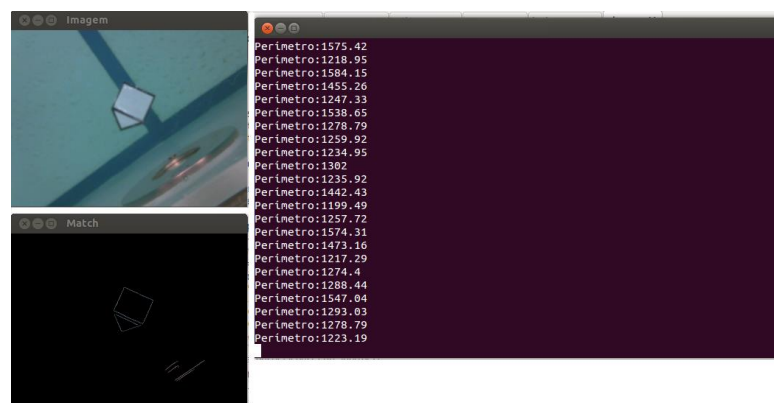
#### 4.3.1.1 Detecção

O alvo a detetar encontra-se submerso e as imagens captadas exibem o alvo, se presente, e pequenas partículas na água. Com isto o conceito principal passa por extrair os contornos dos objetos presentes na imagem. Seguidamente é calculado o somatório dos perímetros dos objetos, uma vez que na ausência do alvo o resultado do somatório varia entre 0 e 800 pixéis. Com a presença do alvo o valor estará acima dos 800 pixéis. Utilizou-se uma margem de 200 pixéis para garantir a ausência de falsos positivos, ou seja, apenas acima dos 1000 pixéis é considerada a presença do alvo. Ambos os valores foram calculados com base em dados obtidos nos vários testes efetuados em cenário próximo do real, com e sem alvo, com o mesmo em diferentes profundidades.



**Figura 4.6** – Resultado do perímetro sem alvo.

Na figura 4.6 é possível visualizar o resultado da ausência do alvo na imagem. Já na figura 4.7 pode-se observar o resultado obtido com a presença do alvo.



**Figura 4.7** – Resultado do perímetro com alvo.

O processamento de cada imagem é efetuado do seguinte modo: a imagem é convertida para a escala de cinzentos; é aplicado um filtro para a desfocar de modo a reduzir o ruído com uma janela de tamanho 3; e por último é aplicada a função *Canny edge detector* para a deteção de contornos da biblioteca OpenCV [22].

Os parâmetros necessários à evocação da função Canny foram obtidos com aplicação deste processo a imagens em ambiente próximo do real e com o auxílio de uma *trackbar*. Esta tem o objetivo de alterar os parâmetros limiar mais baixo e limiar mais elevado de modo a serem afinados de acordo com a necessidade. Assim, foi possível definir os limiares que mais convinham a este caso.

Aplicado o processo acima descrito obteve-se os contornos dos objetos, faltando apenas o calculo do perímetro de cada um e, posteriormente, o somatório de todos, para conclusão da deteção.

Foi possível obter o perímetro de cada objeto com auxílio da função presente na biblioteca OpenCV `findContours()` [23]. Esta função guarda o número de pixéis correspondentes aos contornos de cada objeto num vetor, pelo que foi apenas necessário somar todos os perímetros presentes na imagem, obtendo assim o valor pretendido. As figuras 4.6 e 4.7, anteriormente expostas, apresentam o resultado do processo aqui descrito. A janela “Match” presente nas figuras é o produto final, juntamente com o perímetro.

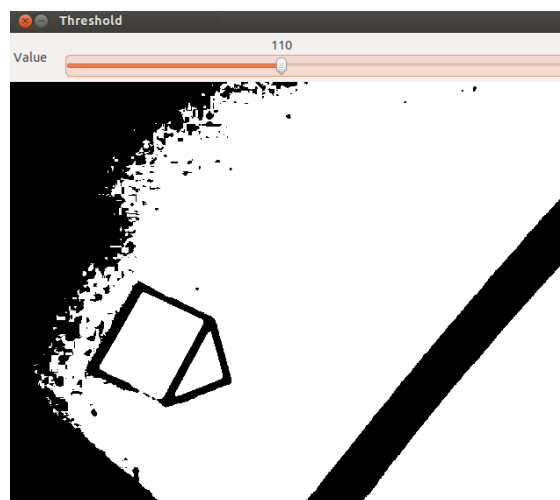
O processo de deteção aqui descrito em termos de processamento computacional é relativamente pesado, uma vez que o objetivo final será implementar numa *raspberry PI* com um *cpu single core* a 700 MHz. Isto porque esta não tem o mesmo nível de processamento de um portátil atual, intel core i5 com *cpu* de 2.3 GHz como o portátil utilizado para desenvolver o código. Por isso teve-se que otimizar este processo. A solução passou por correr este algoritmo de 3 em 3 *frames*. O intervalo de *frames* teve como critérios a velocidade dos veículos na água e o peso computacional. Assim, este intervalo permite aumentar a rapidez de processamento e, simultaneamente, não influencia a deteção do alvo.

#### 4.3.1.2 Análise

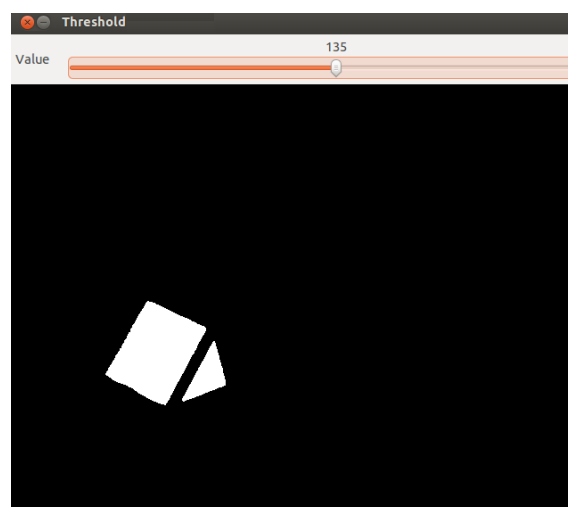
O algoritmo de análise contém duas fases, sendo a primeira a extração das posições x e y em pixéis, relativamente à câmara, e a orientação. A origem final será o centro da câmara subaquática. A segunda fase corresponde à conversão entre as coordenadas da imagem e as coordenadas da câmara e, posteriormente, a passagem a coordenadas mundiais. É de salientar que a orientação é calculada já com as medidas x e y em metros, tendo como unidade de medida o grau.

### Extração de informações

Relativamente à primeira fase, a *frame* depois de captada e de ter passado pelo processo de detecção, tendo este confirmado a presença do alvo, inicia a passagem por vários processos até à extração das informações pretendidas. A imagem é convertida para a escala de cinzentos, dando a formação de uma nova imagem. Esta passa pelo processo de segmentação, sendo que o limiar escolhido tem em consideração o alvo seleccionado. A cor predominante é o branco e, através de várias medições do histograma, foi possível determinar a intensidade dos pixéis pertencentes ao alvo. Com isto adquiriu-se o valor 135 para o limiar. As figuras 4.8 e 4.9 apresentam o processo utilizado para a escolha do limiar.



**Figura 4.8** – Segmentação de imagem com limiar a 110.



**Figura 4.9** – Segmentação de imagem com limiar a 135.

O processo consistiu em captar uma imagem do alvo submerso e com auxílio de uma *trackbar* obter o valor do limiar desejado.

Aplicou-se também as operações de erosão e fecho para precaver a presença de ruído prejudicial à forma do alvo.

A presença de manchas brancas, *blobs*, isoladas juntamente com o objeto de interesse é uma realidade quando se está a lidar com manipulação de imagens. Isto deve-se ao facto de existir pixéis isolados ou pequenos grupos de pixéis com intensidades que estão acima do limiar escolhido. Esta ocorrência prejudica a extração das informações pretendidas, uma vez que as operações precedentes reconhecem o alvo como sendo os pixéis brancos presentes na imagem. A presença destas pequenas manchas brancas poderá levar à aquisição de informações erradas, o que não é o previsto. A solução passou por calcular a área de cada *blob* e seleccionar/apresentar apenas aquela com maior área, considerando esta como sendo o alvo. As funções `findContours()` [23] e `contourArea()` [24] da biblioteca OpenCv foram usadas para obter o resultado desejado. Obtido o alvo calculou-se o ponto correspondente ao seu centro de massa.

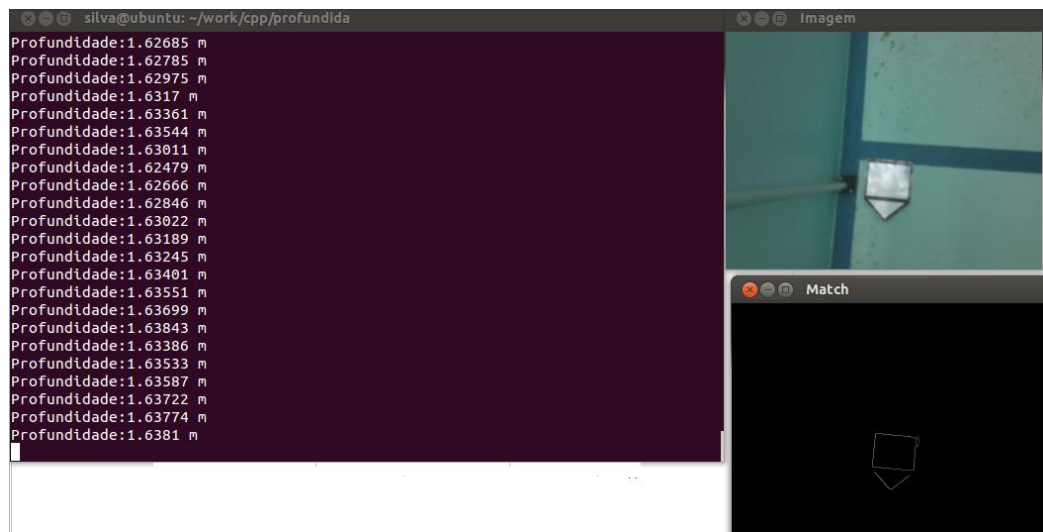
Obter a profundidade do alvo foi essencial, uma vez que este parâmetro é necessário na calibração. Como foi referido no início desta secção não foi possível determinar a profundidade com exatidão devido aos aspetos anteriormente referidos. Contudo, foi possível, apenas com uma câmara, estimar tal parâmetro. O procedimento baseou-se na recolha de medidas do perímetro do alvo, em pixéis, em diferentes profundidades. A tabela 4.2 descreve as medidas efetuadas e os respetivos resultados.

**Tabela 4.2** - Medidas de profundidade.

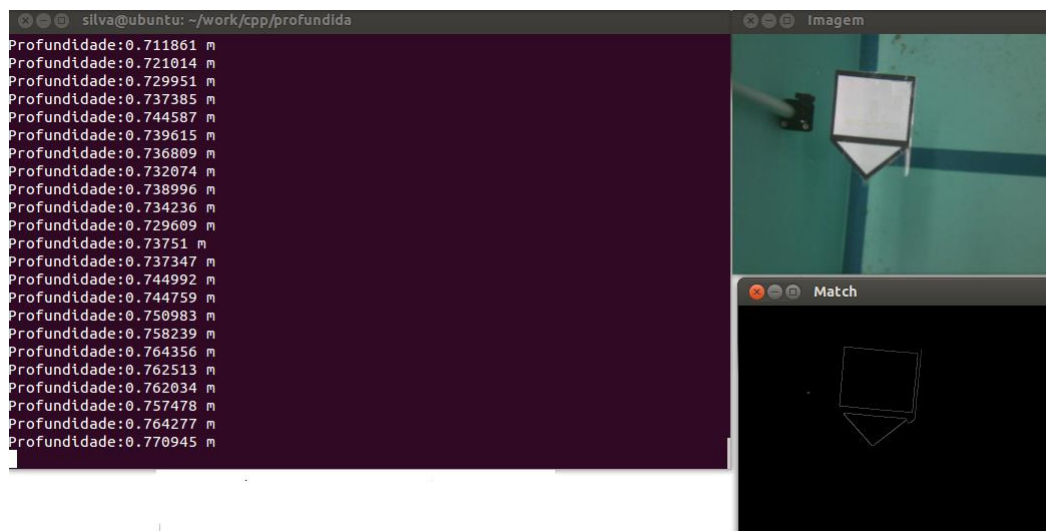
|                    |      |     |     |
|--------------------|------|-----|-----|
| Profundidade (m)   | 0.65 | 1   | 1.7 |
| Perímetro (pixéis) | 1300 | 580 | 430 |

Com isto e com a aproximação por uma reta do método dos mínimos quadrados  $Z=C_1+C_2P$ , onde  $P$  é o perímetro em pixéis e  $Z$  a profundidade em metros, chegou-se aos valores de  $C_1$  e  $C_2$ , obtendo-se a função  $Z = 215.6050 - 0.1350P$ .

Assim, foi possível estimar a profundidade do alvo como se pode verificar nas figuras 4.10 e 4.11.

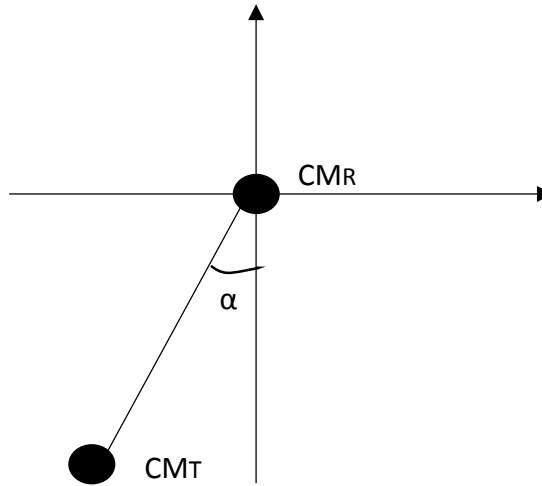


**Figura 4.10** – Estimação da profundidade com alvo a 1.65 m.



**Figura 4.11** – Estimação da profundidade com alvo a 0.8 m.

A orientação foi obtida selecionando as áreas correspondentes ao retângulo e ao triângulo e através dos seus centros de massa calculou-se o ângulo entre estes. A figura 4.12 ilustra o processo de obtenção do ângulo.



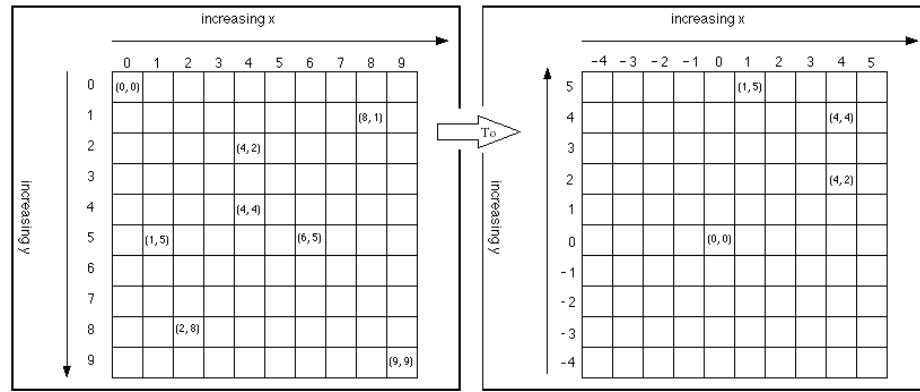
**Figura 4.12** – Ilustração da obtenção da orientação.

### Conversão entre coordenadas

A conversão entre coordenadas é essencial para ser possível localizar o alvo no referencial mundo. Os processos de análise anteriormente referidos obtêm as posições  $x$  e  $y$  no plano da imagem em pixels, tendo como origem o canto superior esquerdo. A conversão foi feita multiplicando a matriz de calibração intrínseca  $M_{in}$  inversa, obtida na calibração, ao ponto correspondente ao centro de massa do alvo,

$$\begin{bmatrix} 656.0040 & 0 & 344.4945 \\ 0 & 656.8646 & 249.9937 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix}. \quad (4.5)$$

O ponto obtido pertence então ao plano da câmara e é mensurável em metros.



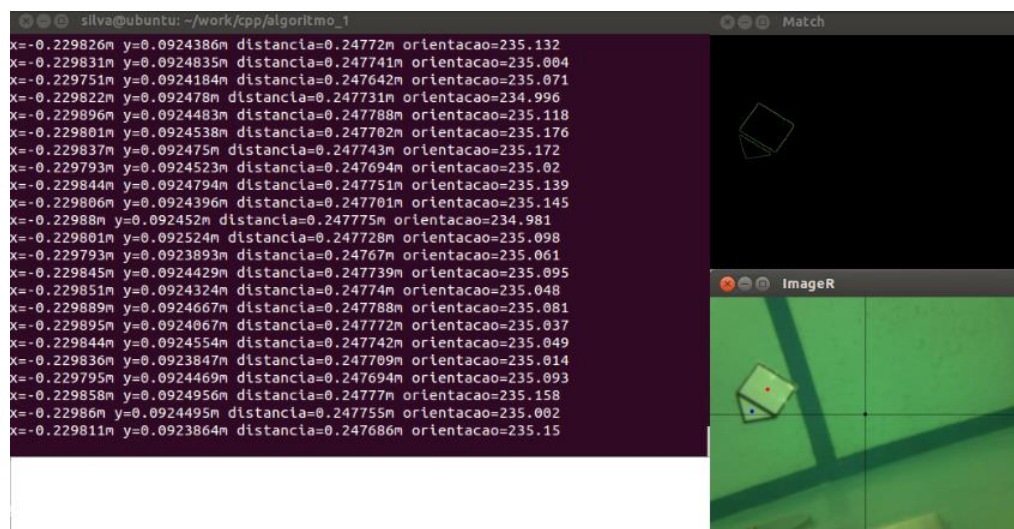
**Figura 4.13** – Ilustração da mudança de origem.

A figura 4.13 ilustra a passagem da origem do canto superior esquerdo para o centro. Os valores apresentados são exemplos, sendo que apenas se pretende salientar a mudança da origem de um plano para outro.

Para finalizar foi apenas necessário aplicar  $f_x$  e  $f_y$  aos pontos  $x_c$  e  $y_c$  respetivamente, para se obter os pontos de interesse nas coordenadas mundo,

$$\begin{bmatrix} x_m \\ y_m \end{bmatrix} = \begin{bmatrix} 0.7309 \cdot z + 0.1722 & 0 \\ 0 & 1.0208 \cdot z - 0.1388 \end{bmatrix} \cdot \begin{bmatrix} x_c \\ y_c \end{bmatrix}. \quad (4.6)$$

Na figura 4.14 é possível verificar o resultado de todo o processo da visão computacional.



**Figura 4.14** – Resultado final do sistema óptico.



### 4.3.2 Acústica

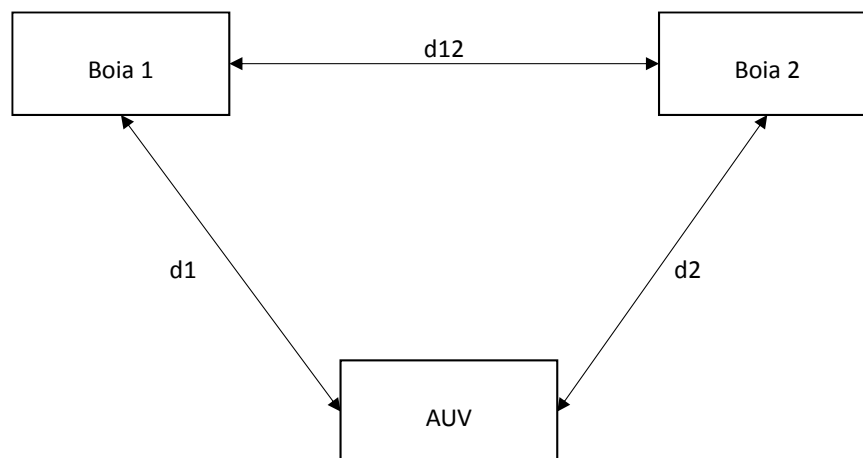
O algoritmo desenvolvido foi criado de forma a aproveitar as particularidades da navegação apoiada em redes acústicas de posicionamento.

Para a operação dos seus veículos autónomos, o OSG tem disponível uma rede acústica composta por vários faróis acústicos que auxilia nas tarefas de localização dos veículos.

No caso específico dos veículos subaquáticos e concretamente do AUV MARES, este está dotado de um transdutor acústico que lhe possibilita trocar informações com a referida rede, permitindo a sua localização num referencial previamente estabelecido.

No funcionamento de uma rede acústica, os veículos devem interrogar cada uma das boias que a compõem, tal como descrito na secção 2.2, sendo possível através da contagem do tempo de resposta extrapolar a distância entre o veículo e cada uma das boias. Conhecendo este processo, e tendo acesso a estas medidas, é então possível para uma entidade exterior estimar a posição do AUV, baseada no registo das comunicações entre todas as entidades.

Na figura 4.15 é possível visualizar um esquema do método descrito.



**Figura 4.15** – Esquema de troca de sinais acústicos

Em cada ciclo de interrogações, durante o qual se admite que a posição do AUV se mantém inalterada, este envia um pedido a cada boia, na forma de sinal acústico, esperando depois pelas suas respostas. Na tabela 4.2 é então possível ver cada um destes ciclos de comunicações. O AUV envia o pedido num canal e recebe a resposta em canais diferentes, de acordo com a boia em questão. O instante em que se admite que se iniciam as comunicações é denominado por  $t_0$  e  $\delta x$  e  $\delta y$  referem-se ao atraso entre a receção de um evento acústico de uma boia e a resposta por ela emitida.

**Tabela 4.3** - Ciclo de comunicações.

|       | Instantes:                     | Descrição:                  |
|-------|--------------------------------|-----------------------------|
|       | $t_0$                          | AUV interroga B1 e B2       |
| $t_1$ | $t_0 + \Delta t_1$             | B1 recebe pedido de AUV     |
| $t_2$ | $t_0 + \Delta t_2$             | B2 recebe pedido de AUV     |
| $t_3$ | $t_0 + \Delta t_1 + \delta x$  | B1 responde a AUV           |
| $t_4$ | $t_0 + \Delta t_2 + \delta y$  | B2 responde a AUV           |
| $t_5$ | $t_0 + 2\Delta t_1 + \delta x$ | AUV recebe a resposta de B1 |
| $t_6$ | $t_0 + 2\Delta t_2 + \delta y$ | AUV recebe a resposta de B2 |

Tendo em atenção esta troca de mensagens, é possível estabelecer uma relação entre os instantes em que cada um dos eventos acústicos acontece. De facto, os instantes de tempo em que cada uma das boias escuta os pedidos do AUV são suficientes para estimar a posição do AUV. Contudo, os temporizadores das boias e AUV não estão sincronizados, por isso é necessário ter em consideração todos os tempos de pedido e resposta.

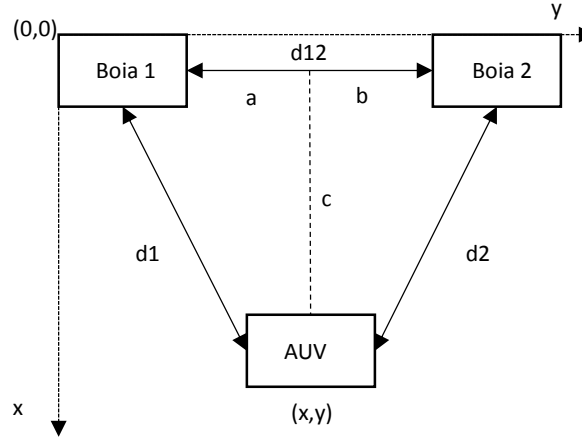
O AUV emite sinais a interrogar as boias, enviando num canal. As boias respondem em canais diferentes. O AUV sabendo inicialmente qual o canal que corresponde a cada boia, determina a distância às mesmas. Tomando como exemplo a distância à boia 1 e utilizando a tabela 4.2, é possível chegar a  $d1$ . Subtraindo o instante de tempo de resposta da boia 1 ao instante de interrogação do AUV fica,

$$t_5 - t_0 = 2\Delta t_1 + \delta x. \quad (4.7)$$

Os intervalos de tempo  $\delta x$  e  $\delta y$  são conhecidos. Sabendo a velocidade de propagação das ondas acústicas na água, é então possível calcular a distância do AUV a B1, já que esta é proporcional a 4.7,

$$d1 = \frac{t_5 - t_0 - \delta x}{2} * v_{som}. \quad (4.8)$$

Obtida a distância a cada boia é então necessário calcular a posição  $x$  e  $y$  do veículo em relação às boias.



**Figura 4.16** - Ilustração do cálculo da posição.

Com recurso á ilustração da figura 4.16, pode-se explicar como se obtém x e y do AUV, tendo como incógnitas a, b e c obtém-se o seguinte sistema de equações:

$$\begin{cases} d_{12} = a + b \\ d_1^2 = a^2 + c^2 \\ d_2^2 = b^2 + c^2 \end{cases} \quad (4.9)$$

Desenvolvendo o sistema obtém-se as seguintes expressões:

$$a = \frac{d_1^2 - d_2^2 + d_{12}^2}{2 * d_{12}}, \quad (4.10)$$

$$b = \sqrt{\frac{d_1^2 - d_2^2 - d_{12}^2 + 2 * d_{12} * d_2^2}{2 * d_{12}}}, \quad (4.11)$$

repare-se que  $\underline{a}$  é a posição em  $\underline{y}$  e  $\underline{c}$  a posição em  $\underline{x}$ .

### 4.3.3 Fusão e estimação

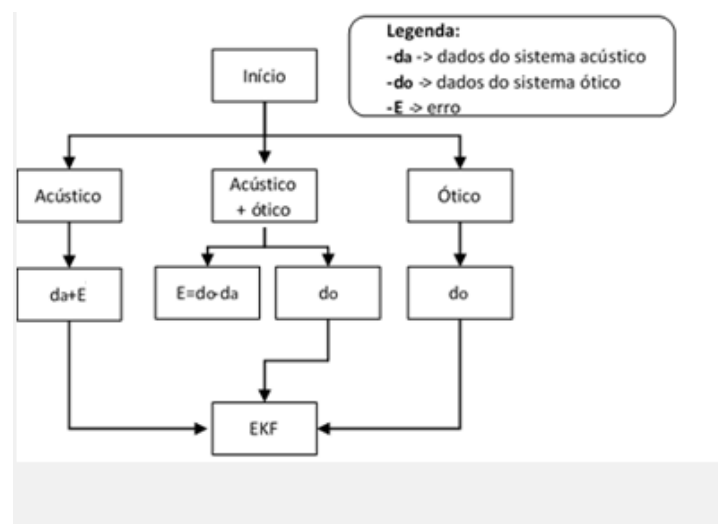
Nas secções anteriores foram descritos os processos de obtenção das posições do alvo relativamente ao ASV. No entanto, o objetivo final é conjugar essas informações de posição do alvo, de modo a melhorar o seguimento deste por parte do ASV.

Os métodos de atualização são processos discretos tendo taxas de atualização de 1Hz no caso acústico e no caso óptico de 16Hz, se for processado pelo portátil ou 2Hz se pelo *raspberry PI*. No entanto, é necessário estabelecer continuamente a posição do alvo para determinar corretamente a sua trajetória.

### Filtro de Kalman

Usando um filtro de Kalman contínuo-discreto é possível prever a evolução do sistema, corrigindo estas evoluções sempre que se dispõe de novas medidas de posição.

As medidas de posição antes de atualizarem o filtro têm que ser fundidas de modo a tirar partido de ambos os métodos. Assim sendo, construiu-se a fusão do seguinte modo: quando existem dados de ambos os métodos calcula-se o erro entre as posições da acústica e as do óptico, atualizando de seguida o filtro apenas com os dados provenientes do método óptico. No caso de apenas existir dados do método acústico é adicionado o erro anteriormente calculado, atualizando de seguida o filtro. A abordagem mencionada tem como objetivo garantir que se utilize os dados mais precisos. A figura 4.17 ilustra o processo descrito.



**Figura 4.17** – Ilustração da fusão de dados.

Tal como descrito em [2], e considerando que durante as missões o alvo (AUV) se move fundamentalmente ao longo de linhas retas, é admissível modelar a dinâmica do AUV do seguinte modo:

$$\begin{bmatrix} \dot{x} \\ \dot{v}_x \\ \dot{y} \\ \dot{v}_y \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \beta & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \beta \end{bmatrix} \begin{bmatrix} x \\ v_x \\ y \\ v_y \end{bmatrix} + \begin{bmatrix} n_x \\ n_{vx} \\ n_y \\ n_{vy} \end{bmatrix}. \quad (4.12)$$

Em 4.12 (x,y) referem-se às coordenadas do veículo no plano horizontal, enquanto (v<sub>x</sub>,v<sub>y</sub>) se referem às velocidades. O parâmetro  $\beta$  representa um decaimento exponencial da velocidade, e encontra justificação na necessidade que as duas componentes da velocidade tendam para zero, se, por qualquer motivo, não chegarem novas medidas. Desta forma, garante-se que no limite o AUV irá parar. O vetor  $(n_x; n_{vx}; n_y; n_{vy})^T$  modeliza as diferenças entre o movimento real do veículo e o movimento previsto por este modelo. Usado um filtro contínuo-discreto de Kalman é então possível integrar as medidas de posição obtidas através dos métodos referidos com uma previsão do comportamento do sistema [25].

Sempre que se dispõe de uma nova medida, atualizam-se as matrizes de estado e covariância,  $X_k$  e  $P_k$ , para novos valores,

$$X_k = X_k^- + K(z_k - z_k^*), \quad (4.13)$$

$$P_k = (I - K_k H)P_k^-, \quad (4.14)$$

$z_k$  identifica a nova medida de posição, e  $z_k^*$  identifica a previsão dada pelo filtro de Kalman para essa mesma posição.

A matriz de ganho  $K_k$  é calculada tal como indicado em 4.15 e  $r_s$  representa a covariância do erro associado à medida de  $z_k$ ,

$$K_k = P_k^- H^T S_k^{-1}, \quad (4.15)$$

$$S_k = H P_k^- H^T + r_s. \quad (4.16)$$

A estimação da dinâmica não é completamente satisfatória. De facto, apesar de as estimativas da posição (x,y) serem bastante fiáveis, o mesmo não acontece para as estimativas da velocidade (v<sub>x</sub>, v<sub>y</sub>). O mesmo quer dizer que, tal como o modelo 4.12, não fornece grande informação em relação ao comportamento da velocidade para além de prever o seu decaimento exponencial, as várias estimativas produzidas contém pouca informação relevante.

### Estimador recursivo do método de mínimos quadrados (RLS)

Como anteriormente referido, o movimento do AUV é feito em linhas retas, pelo que as equações que regem o movimento, podem ser descritas como,

$$x(t) = x_0 + v_x t, \quad (4.17)$$

$$y(t) = y_0 + v_y t. \quad (4.18)$$

Em 4.17 e 4.18 está descrito o movimento segundo as componentes x e y, respetivamente, sendo  $(x_0, y_0)$  as posições no plano horizontal para o instante  $t=0$  e  $(v_x, v_y)$  as velocidades. Estas equações são lineares e podem ser utilizadas para com o auxílio de um algoritmo de mínimos quadrados obter melhores estimativas que as fornecidas pelo filtro de Kalman para as velocidades do AUV no plano horizontal.

O algoritmo dos mínimos quadrados recursivo passa pelo cálculo da matriz de ganho, dos parâmetros e da covariância do erro, da seguinte forma,

$$K_n = \frac{P_{n-1} \varphi_n}{1 + \varphi_n^T P_{n-1} \varphi_n}, \quad (4.19)$$

$$\theta_n = \theta_{n-1} + K_n [y_n - \varphi_n^T \theta_{n-1}], \quad (4.20)$$

$$P_n = [I - K_n \varphi_n^T \theta_n^T] P_{n-1}. \quad (4.21)$$

A partir de 4.19, 4.20 e 4.21 estima-se as posições  $(x_0, y_0)$  e as velocidades  $(v_x, v_y)$  tendo como dados as posições  $(x, y)$  e o instante em que ocorrem,  $t_i$ .



## Capítulo 5

# Implementação

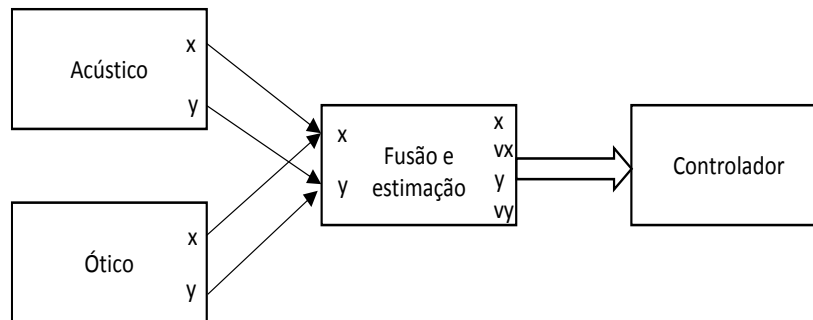
Até ao momento foi descrito e referenciado todo o trabalho desenvolvido, necessário à construção do sistema final. Neste capítulo será descrita a implementação desses subsistemas de modo a ser criado, como produto final, o sistema proposto nos objetivos. Este consistiu num *software* desenvolvido com o intuito de ser inserido numa *raspberry PI* e, por sua vez, adicionado por modo *plug&play* a uma embarcação autónoma.

Como já salientado ao longo da dissertação, o *software* foi desenvolvido em C/C++ e compilado em ambiente Linux. A escolha deveu-se ao facto de o *software* utilizado na OSG ser maioritariamente em C++ e ser o sistema operativo utilizado pela *raspberry PI*.

O programa consiste em três processos a correr simultaneamente, cada um correspondente a cada subsistema, visão, acústica e fusão e estimação. A arquitetura utilizada é do tipo servidor/cliente. A escolha deste tipo de arquitetura prende-se com o facto de existir dois subsistemas independentes na aquisição de dados, óptico e acústico, e um subsistema que funde esses dados com o intuito de informar o controlador. Sendo o último o centro de toda a operação foi escolhido para servidor, ou seja, os clientes comunicam com o servidor.

O diagrama da figura 5.1 exemplifica a troca de informações entre os processos.





**Figura 5.1** – Diagrama de interação entre processos

O modelo comunicações entre processos baseou-se em *sockets* UDP com protocolo TCP/IP. Na figura 5.2 é possível ver um excerto de código do modo de comunicação entre clientes e servidor.

```


Cliente


char message[BUFLen];
struct sockaddr_in si_other;
int s, i, slen=sizeof(si_other);
//Create socket
s=socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP);
memset((char *) &si_other, 0, sizeof(si_other));
//send the message
sendto(s, message, strlen(message), 0, (struct sockaddr *) &si_other, slen);



Servidor

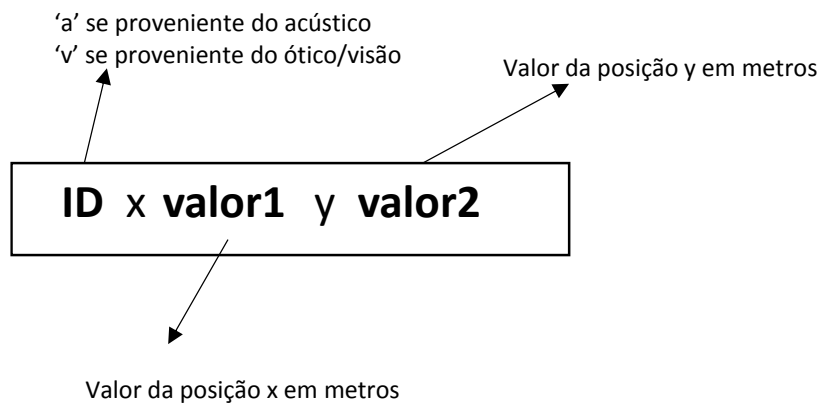

struct sockaddr_in si_me, si_other;
int s, i, rcv_len;
socklen_t slen = sizeof(si_other);
char buf[BUFLen];
//create a UDP socket
s=socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP);
//bind socket to port
bind(s, (struct sockaddr*)&si_me, sizeof(si_me));
//try to receive some data
rcv_len = recvfrom(s, buf, BUFLen, 0, (struct sockaddr *) &si_other, &slen);

```

**Figura 5.2** – Excerto do código para comunicação entre clientes e servidor.

O servidor fica à “escuta” num porto predefinido e sempre que cada cliente tem dados, envia-os para esse porto de modo a serem capturados e processados pelo servidor.

A estrutura de dados enviados pela trama UDP é do tipo *string*. Todos os dados produzidos pelos clientes são inicialmente do tipo *double*, sendo convertidos aquando do envio para um formato predefinido do tipo *string*. O formato utilizado pode ser visto na figura 5.3.



**Figura 5.3** – Formato da string.

O processo relativo ao subsistema acústico, como referenciado em capítulos anteriores, necessita de informações do repositório. O acesso ao repositório é feito através de uma requisição/registo no centralizador de informação, baseando-se este numa arquitetura do tipo modelo centralizado. Regista-se no centralizador e é feito um pedido das informações que se quer receber. Os pacotes de dados são enviados no momento que chegam ao centralizador. O registo é feito através de *socket* UDP e a mensagem é do tipo estrutura com campos específicos que são preenchidos de acordo a informação pretendida. A figura 5.4 mostra um exemplo deste registo.

```

//*****Variables*****
IPMsg msg;
int s;
struct sockaddr_in addr;
//*****Inicializations*****
msg.header.id = 0; // irrelevante
msg.header.subid = 0; // irrelevante
msg.header.msgtype = STA_GET_MSGS; // pedido de envio de dados
msg.msg.stagetmsgs.id = 0; // 0: todos os sistemas
msg.msg.stagetmsgs.subid = 0; // 0: todos os subsistemas
msg.msg.stagetmsgs.msgtype = 0; // 0: todos os tipos de dados
msg.msg.stagetmsgs.rate = 0; // 0: máxima taxa

```

**Figura 5.4** – Excerto de código para registo no centralizador.

É de salientar que a OSG tem um protocolo de comunicações implementado nos seus sistemas e, para tal, tem disponível um ficheiro .h com as estruturas definidas. Esse ficheiro foi utilizado neste projeto de modo a possibilitar a comunicação com as boias e o repositório. A IPMsg é uma dessas estruturas.

O modo de comunicação com as boias é feito também através de *socket* UDP. Foi necessário a configuração destas com o objetivo de definir as frequências de receção e de emissão das ondas

acústicas. Por fim é apenas necessário calcular as distâncias e as posições  $x$  e  $y$  como referido em 4.3.2, para posteriormente enviar para o servidor.

O processo responsável pelo algoritmo correspondente à parte de visão computacional ordena a captura da imagem através do comando VideoCapture cap(1), disponível na biblioteca OpenCV. Essa imagem passa pelas várias fases do algoritmo e, no final, são extraídas informações de posicionamento, sendo estas enviadas posteriormente para o servidor.

No servidor são reunidas e fundidas todas as informações provenientes dos processos acústico e óptico. Este quando recebe uma trama, descodifica a mensagem de modo a conhecer os valores de  $x$  e  $y$ , e a fonte das informações, acústica ou ótica. O algoritmo desenvolvido está preparado para de acordo com a fonte proceder de maneiras diferentes. Se a fonte for o processo óptico atualiza-se o filtro de Kalman com os valores recebidos. Aquando da receção das posições provenientes do acústico calcula-se o erro entre as medidas acústicas e óticas. Sabendo que não se irá receber mais medidas óticas num intervalo de tempo estipulado adiciona-se o erro às posições do acústico e, de seguida, atualiza-se o filtro. O intervalo de tempo entre amostras é medido no servidor, uma vez que os tempos de processamento e envio de dados são constantes, não influenciando o resultado. Com estes intervalos é possível obter o tempo de cada amostra, como se demonstra, de seguida. Considere-se  $t_{1r}$  e  $t_{2r}$  os tempos precisos de duas amostras,  $t_{1o}$  e  $t_{2o}$  os tempos obtidos dessas amostras e  $E$  os erros temporais relativos à aquisição, processamento e envio das amostras, sendo estes constantes, então,

$$t_{1o} = t_{1r} + E, \quad (5.1)$$

$$t_{2o} = t_{2r} + E. \quad (5.2)$$

Ao calcular o intervalo de tempo entre amostras,  $\Delta t$ , anula-se os erros,

$$\Delta t = t_{2o} - t_{1o} = t_{2r} + E - t_{1r} - E. \quad (5.3)$$

O código do temporizador utilizado encontra-se na figura 5.5.

```
std::clock_t start;
start = std::clock();
double duration, tmp=atof(argv[1]);
duration = (( std::clock() - start ) / (double) CLOCKS_PER_SEC)*1000;
```

**Figura 5.5** – Excerto de código do temporizador.

Os valores  $x$  e  $y$ , calculados pelo filtro, são passados, assim como o tempo de cada amostra ao estimador recursivo de mínimos quadrados. No final do processo de fusão e estimação obtêm-se os dados necessários ao seguimento,  $x$ ,  $y$ ,  $v_x$  e  $v_y$ .

O controlador necessita das posições em latitude e longitude, coordenadas geográficas. Assim, por fim, recorreu-se a bibliotecas de conversão de coordenadas em metros para latitude e longitude em graus.



## Capítulo 6

# Resultados

Neste capítulo são apresentados os testes efetuados ao sistema desenvolvido. Testou-se o *software* desenvolvido com os três algoritmos, visão, acústico e fusão e estimação, para com isto ser possível avaliar o comportamento do sistema em ambiente real. Este capítulo divide-se em descrição do ambiente e ferramentas e de acordo com os vários testes efetuados.

### 6.1 Ambiente e ferramentas

O sistema foi testado utilizando a piscina do departamento de robótica da FEUP. Utilizou-se três transdutores, dois a funcionar como boias acústicas e um a funcionar como alvo, sendo acoplado ao alvo desenvolvido em [4.2](#). De modo a simular o ASV, este foi composto pelo portátil, a correr os algoritmos em ambiente Linux, e pela câmara subaquática.

Criou-se uma área de trabalho limitada pelas boias e obviamente pelo tamanho da piscina, consistindo, respetivamente, pelos 2.8m de largura dados pela distância entre as boias e pelos 4.5m de comprimento da piscina. Como não houve necessidade de movimentar o suposto ASV, fixou-se este na posição  $x=3.45m$  e  $y=0.72m$  em relação às boias.



**Figura 6.1** – Ambiente de testes.

A figura 6.1 apresenta uma foto captada no ambiente de trabalho.

## 6.2 Teste nº1

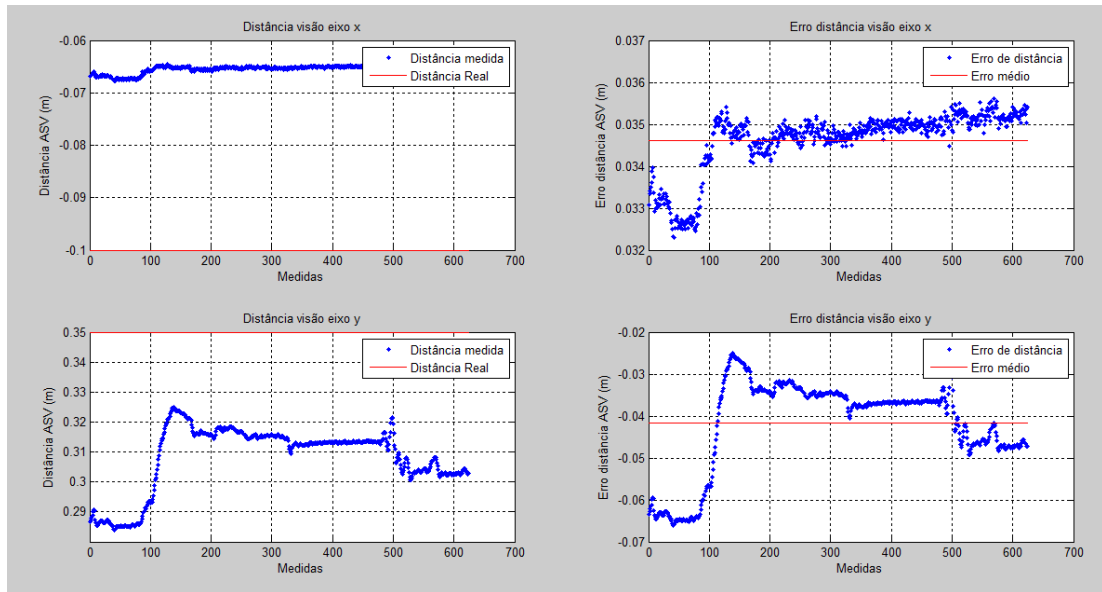
O primeiro teste consistiu em avaliar o erro de medida dos subsistemas acústico e óptico separadamente. Assim, posicionou-se o ASV numa posição conhecida dentro da área de trabalho. De seguida, colocou-se também o alvo numa posição conhecida, diferente da do ASV. Estabelecidas as posições, mediu-se a distância em x e y em relação ao ASV, utilizando apenas um subsistema. Repetiu-se o teste para duas posições, alternando entre subsistemas.

A tabela 6.1 apresenta as posições testadas em relação ao ASV.

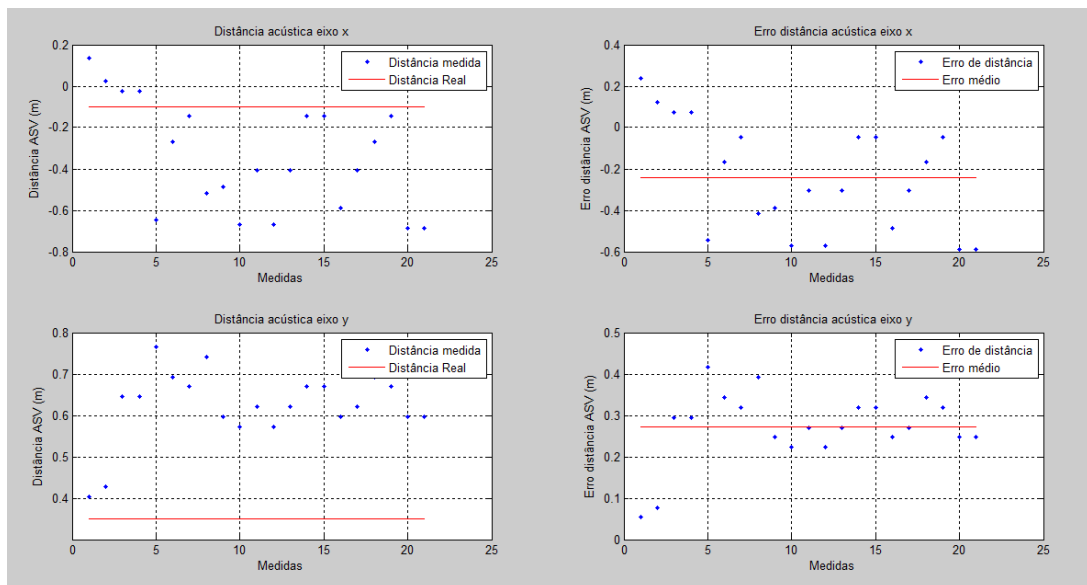
**Tabela 6.1** - Posições testadas.

| Posições: | x (m): | y (m): |
|-----------|--------|--------|
| Posição 1 | -0.1   | 0.35   |
| Posição 2 | -0.1   | 0.25   |

Concluído o teste para a primeira posição, obteve-se os resultados para cada subsistema, presentes nas figuras 6.2 e 6.3.



**Figura 6.2** - Resultado do subsistema óptico para a posição 1.

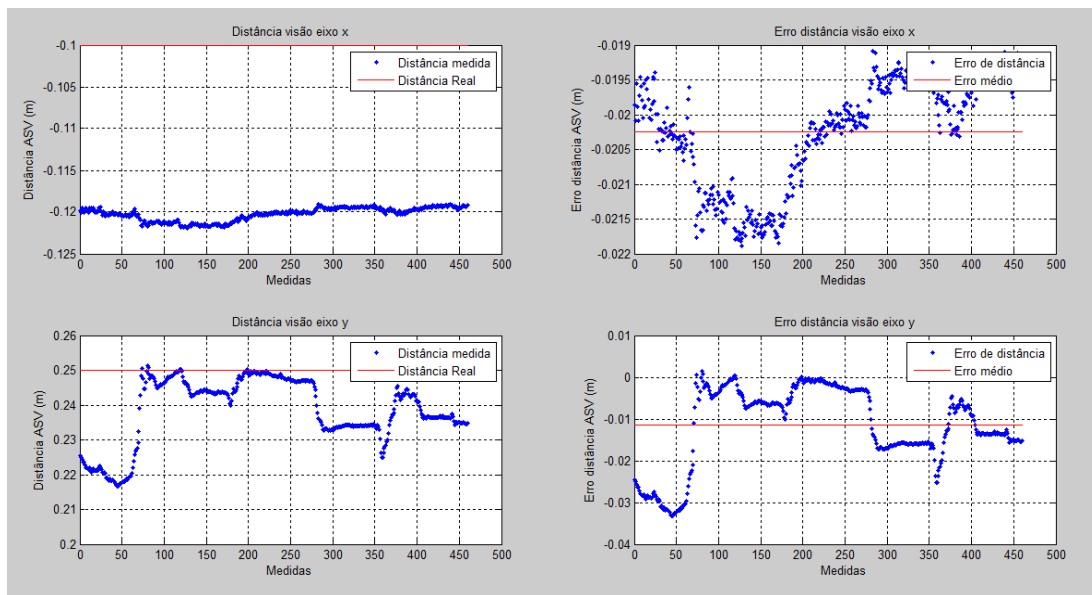


**Figura 6.3** - Resultado do subsistema acústico para a posição 1.

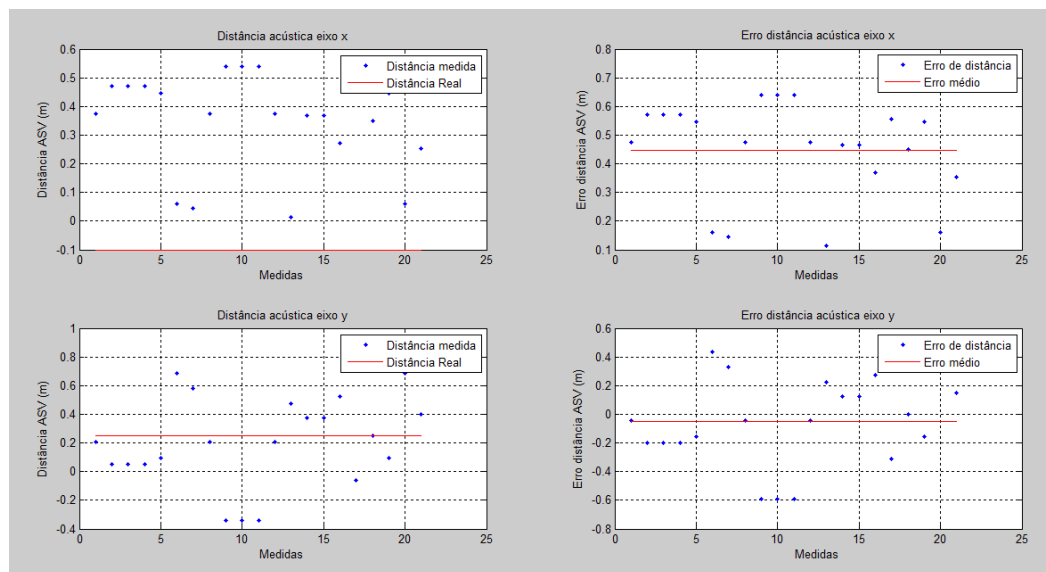
Decidiu-se apresentar os quatro gráficos, pois considerou-se pertinente verificar o erro de cada amostra em cada eixo e o respetivo erro médio. Para a primeira posição é possível concluir, sem qualquer dúvida, que o subsistema óptico comporta-se melhor em termos de erro do que o subsistema acústico, como já era esperado. Além de apresentar um erro médio inferior, em ambos os eixos, ao do acústico, também todas as suas amostras se concentram em torno deste valor,



baixo desvio padrão, ao contrário do subsistema acústico. O desvio padrão em ambos os eixos no acústico é de,  $x=0.2643$  e  $y=0.0860$ , e no caso óptico é de,  $x=7.5023e^{-4}$  e  $y=0.0109$ .



**Figura 6.4** – Resultado do subsistema de visão para a posição 2.

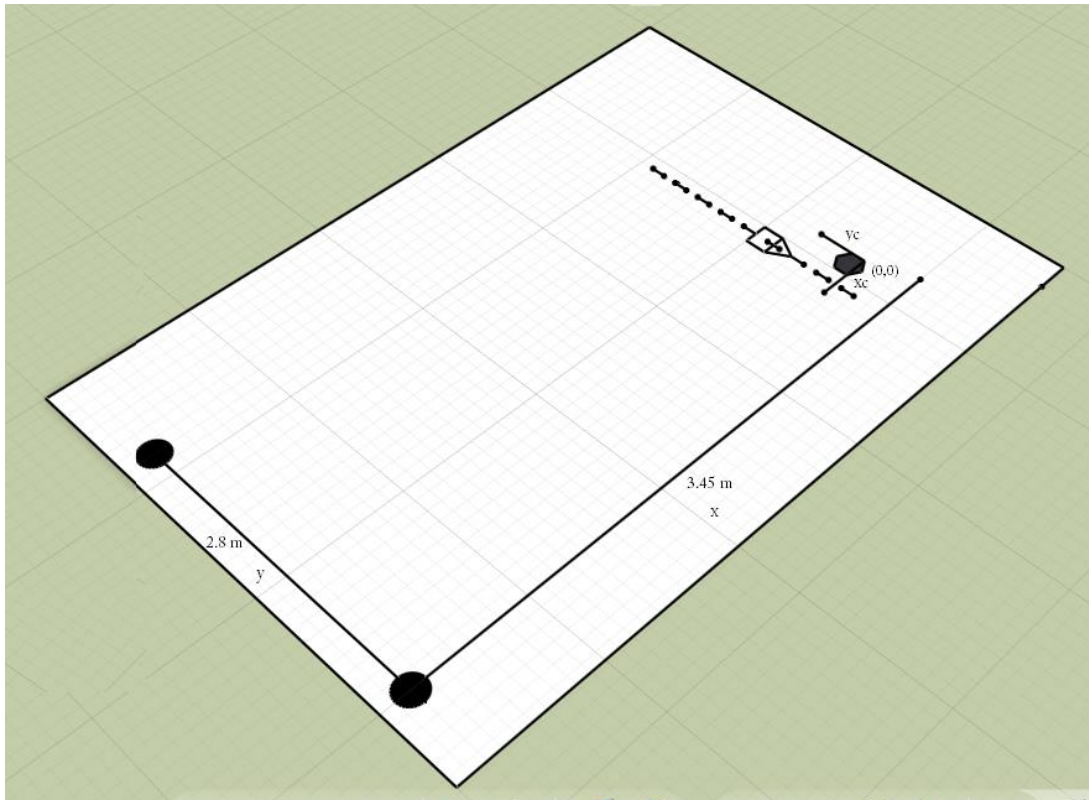


**Figura 6.5** – Resultado do subsistema acústico para a posição 2.

Mais uma vez verificou-se maior precisão do óptico em relação ao acústico. Porém, é preciso analisar uma situação presente no erro médio do eixo y no caso acústico. O erro médio é muito próximo do óptico, o que não é esperado. Com o auxílio das amostras recolhidas é possível justificar tal valor. Esse valor deve-se à elevada variabilidade das amostras e ao facto de os picos serem simétricos, em relação à linha média, fazendo com que estes se anulem, baixando assim o erro.

### 6.3 Teste nº2

O segundo teste teve como objetivo avaliar o funcionamento do sistema completo, ou seja, verificar como este se comporta numa missão de seguimento.



**Figura 6.6** – Esquema do teste realizado.

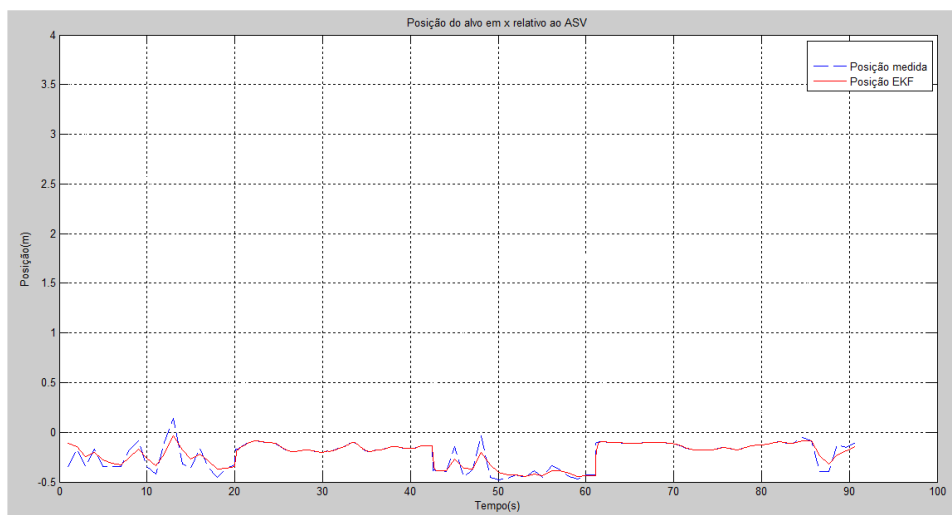


**Figura 6.7** – Foto da realização do teste.

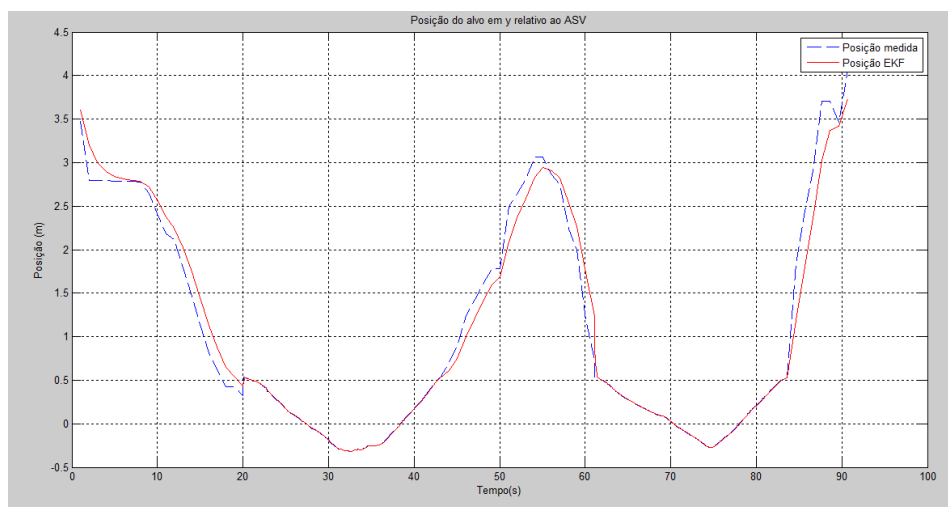
A figura 6.6 representa um esquema do teste realizado. O retângulo presente representa a piscina. As duas circunferências representam as boias acústicas e as linhas x e y representam os eixos do plano das boias. O hexágono consiste na representação do ASV (câmara e portátil), os eixos  $x_c$  e  $y_c$  do plano do ASV são representados pelas linhas correspondentes e a sua interseção representa a origem deste plano. O ASV posiciona-se na posição (3.40, 0.72) relativo ao plano das boias.

Com o ASV fixo movimentou-se o alvo em linha reta, representada pela linha a tracejado. Iniciou-se o teste numa posição distante do ASV e aproximou-se deste. Efetuou-se duas aproximações e dois afastamentos. É de salientar que a posição do alvo é sempre em relação ao plano do ASV.

Na figura 6.7 estão presentes quase todos os elementos necessários à realização do teste, não estando apenas presentes o portátil e as boias.



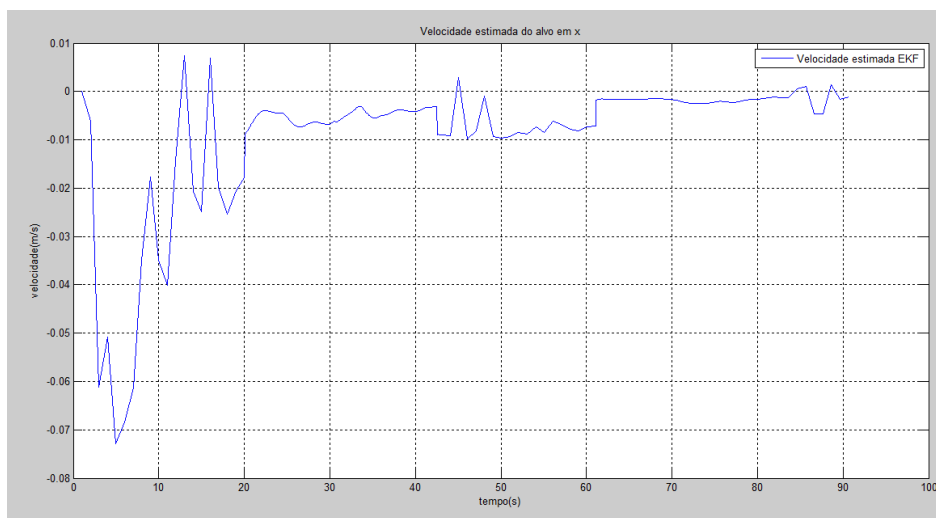
**Figura 6.9** - Posição obtida em x.



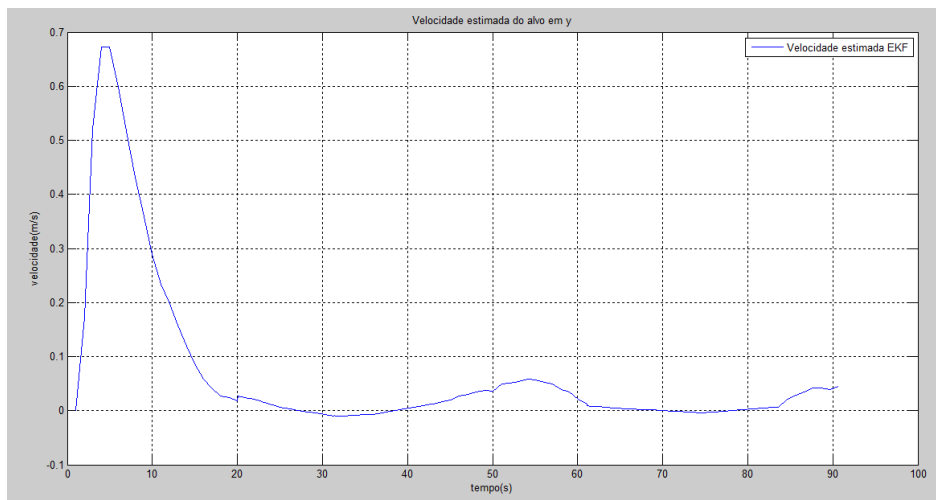
**Figura 6.8** – Posição obtida em y.

Nas figuras 6.8 e 6.9 é possível verificar o resultado do teste, a convergência do filtro de Kalman e a respetiva filtragem em ambos os eixos. É também possível ver a ocorrência de algum ruído, sendo que tal situação deve-se à imprecisão do subsistema acústico. O movimento em x foi feito de modo a que a sua posição se mantivesse constante. Na verdade tal não aconteceu, pois a resistência da água e o movimento manual levaram a pequenas variações.

No caso da velocidade em cada componente apenas se tem a estimativa do filtro de Kalman. Como é possível verificar nos gráficos das figuras 6.10 e 6.11, não existe informação sobre a velocidade, pelo que apenas está presente a sua estimativa. Contudo, sabe-se que a velocidade real estará entre 0.1 m/s e 0.2 m/s.



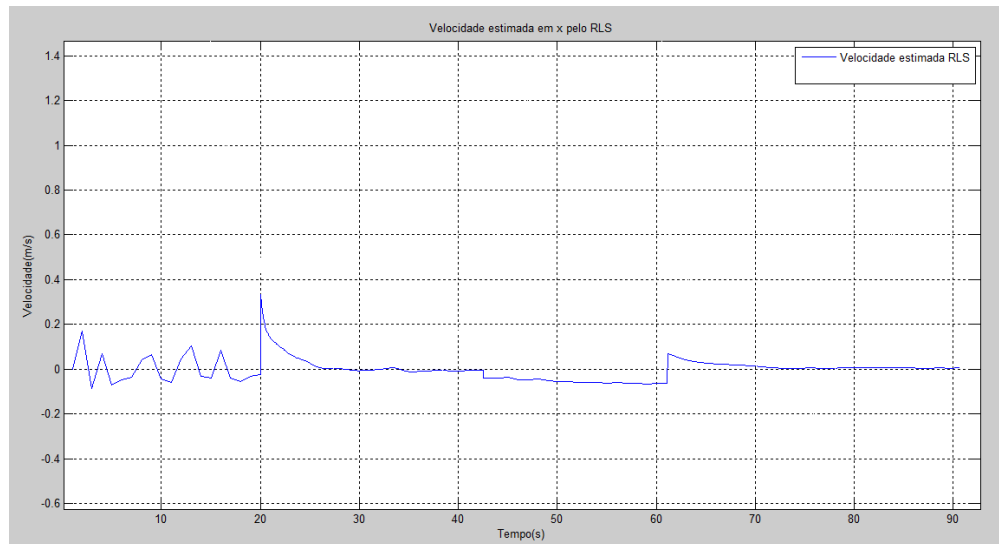
**Figura 6.10** – Velocidade estimada pelo EKF em x.



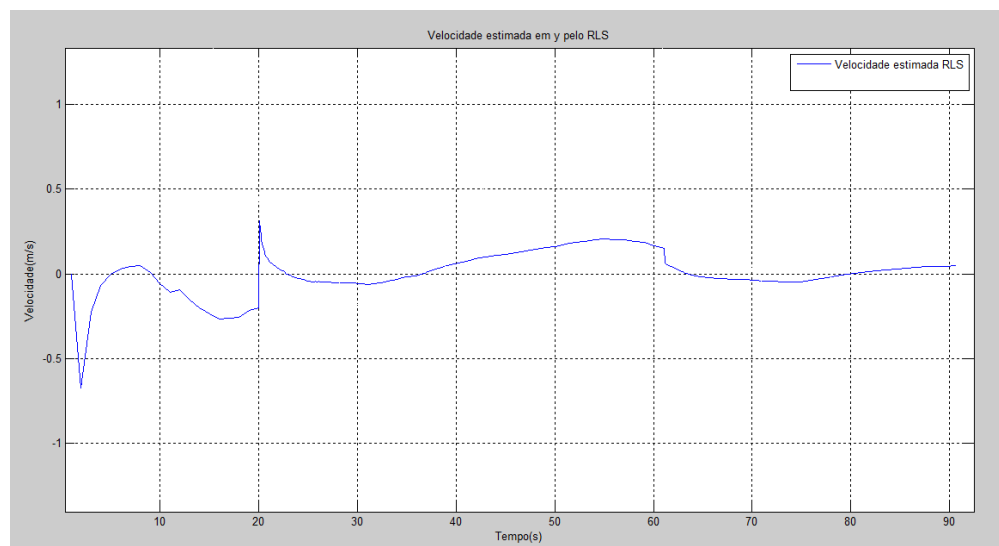
**Figura 6.11** – Velocidade estimada pelo EKF em y.

Analisando as figuras e como previsto em 4.3.3, verifica-se que a informação retirada de ambas não é de todo satisfatória.

Estimando a velocidade através do método recursivo dos mínimos quadrados (RLS) foi possível obter os gráficos das figuras 6.12 e 6.13, para os eixos x e y, respetivamente.



**Figura 6.12** - Velocidade estimada pelo RLS em x.



**Figura 6.13** - Velocidade estimada pelo RLS em y.

Através da comparação entre os métodos EKF e RLS para a estimação da velocidade conclui-se que o último tem resultados mais satisfatórios do que o primeiro, uma vez que também contém mais informação.



## Capítulo 7

# Conclusões e Trabalho Futuro

Neste Capítulo é apresentada uma conclusão do projeto realizado e algumas sugestões para trabalho futuro de forma a melhorar e alargar este.

### 7.1 Conclusão

Neste projeto foi elaborado um estudo e implementado um sistema de seguimento utilizando dois subsistemas de recolha de informação. Este sistema foi composto por algoritmos de aquisição de informações de posicionamento acústico e óptico e por um de fusão e estimação de posição e velocidade.

Neste momento é possível concluir que os objetivos propostos foram atingidos de forma satisfatória. Conseguiu-se desenvolver e implementar os três subsistemas necessários ao funcionamento do sistema de seguimento. O sistema final foi também testado em condições próximas do real. Inicialmente definiu-se testar num ambiente real, por exemplo, no oceano, com o ASV a seguir o AUV, mas tal não foi possível. Isto porque envolvia um grande esforço logístico, já que era preciso pedir autorizações e transportar para o local dos testes não só os veículos, como também as boias e todos os equipamentos de apoio necessários. Os testes em ambiente real são também bastante demorados, já que, para além de todo o tempo despendido na preparação e transporte do equipamento, uma vez no local é necessário fazer uma série de procedimentos com vista à calibração dos vários equipamentos. Contudo, conseguiu-se preparar um ambiente de testes, o mais próximo possível do real, com os meios disponíveis.

Os resultados dos testes efetuados, presente em 6, foram bastante satisfatórios. Foi possível analisar o nível de precisão dos subsistemas óptico e acústico, ficando claro que o de visão é uma mais-valia em missões de seguimento de alvos submersos, pois é bastante preciso. Relativamente

ao teste do sistema completo ficou verificado que os dois subsistemas colmatam as desvantagens um do outro, sendo possível obter resultados satisfatórios. O algoritmo de fusão e estimação mostrou ser capaz de usar inteligente e eficazmente as informações proveniente dos dois subsistemas supracitados. Os algoritmos desenvolvidos tiveram sempre em consideração a sua inserção na *rasperry PI*, e com isto teve-se que ter sempre em atenção o nível de processamento exigido. Desta forma, sempre que um algoritmo era desenvolvido era testado na *rasperry PI* para avaliar o seu desempenho.

## 7.2 Trabalho Futuro

Apesar de todo o trabalho desenvolvido, com resultados francamente positivos, este não termina aqui.

Numa perspetiva a curto prazo, o teste do sistema em ambiente real, é sem dúvida o próximo passo a dar pois valorizaria bastante o trabalho desenvolvido. No entanto, num futuro mais distante, há com certeza vários caminhos que podem ser percorridos, e que estarão muito dependentes das áreas de investigação a seguir pelo OSG no futuro.

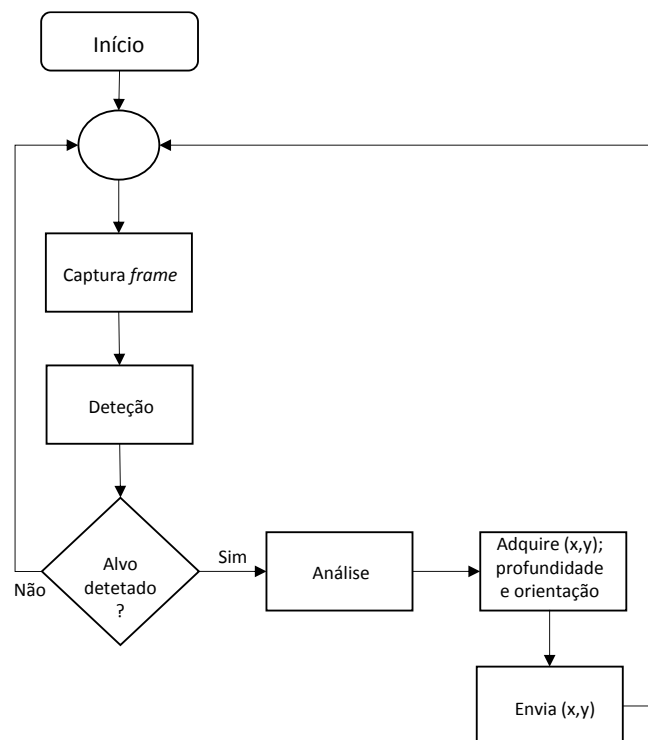
Se porventura se pretender um seguimento da trajetória com menos atraso, mais instantânea, será interessante pensar em formas alternativas de estimar a posição do AUV e que passarão, por exemplo, pela deteção de sinais acústicos pelo próprio ASV.



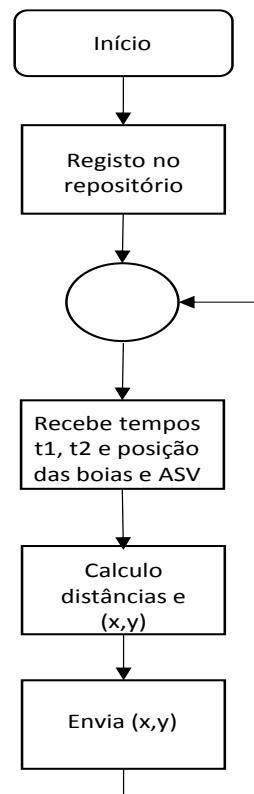
## Anexo A

## Anexos

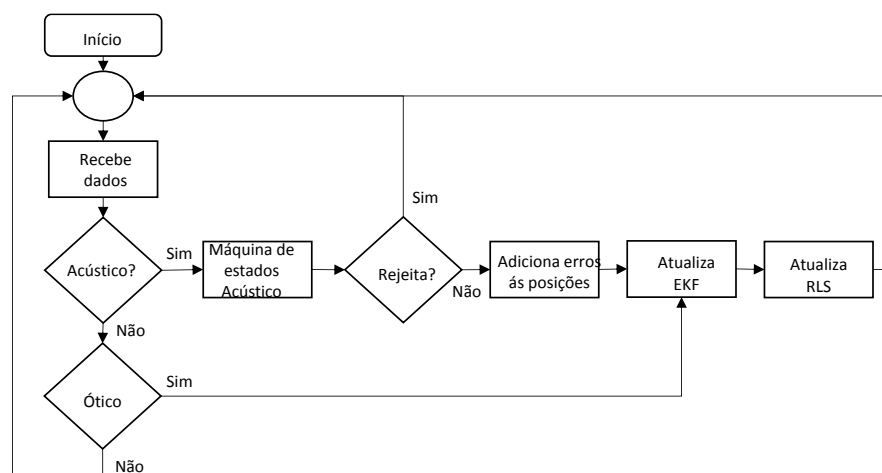
### A.1 Fluxogramas



**Figura A.1** - Processo Óptico.

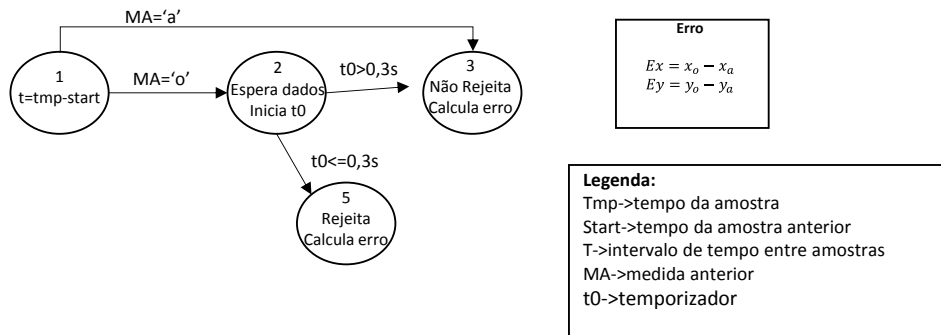


**Figura A.2** - Processo Acústico.



**Figura A.3** - Processo de Fusão e Estimação.

## A.2 Máquinas de Estados



**Figura A.4** - Processo Acústico.

# Referências

- [1] Thor I. Fossen Tristan Perez. Kinematic models for seakeeping and manoeuvring of marine vessels. Modeling, Identification and Control. 2007, 1–12.
- [2] Matos A. and Cruz N. Coordinated Operation of Autonomous Underwater and Surface Vehicles. 2007.
- [3] Daxiong J. A tracking control method of ASV following AUV. S.d.,1-2.
- [4] Rowan E. LBL Underwater Positioning. Lemmer: Hydro International; 2008. Retirado de:[http://www.hydro-international.com/issues/articles/id872-LBL\\_Underwater\\_Positioning.html](http://www.hydro-international.com/issues/articles/id872-LBL_Underwater_Positioning.html).
- [5] Awcock G.W. and Thomas R. Applied Image Processing. McGraw-Hill International Editions. 1996.
- [6] Balihar D. What is a pinhole camera? 2013. Retirado de: <http://www.pinhole.cz/en/pinholecameras/whatis.html>.
- [7] Jepson A. Image Projection. 2011. Retirado de: <http://www.cs.toronto.edu/~jepson/csc420/notes/imageProjection.pdf>.
- [8] Gonzalez, Rafael C. Digital image processing. New Jersey: Prentice Hall. 2002.
- [9] Sonka M, Hlavac V, Boyle R. Image processing, analysis and machine vision. Vitoria [etc.]: Thomson. 2008.
- [10] Jähne B. Digital Image Processing. Springer-Verlag, 2002.
- [11] Acharya T. and Ray A. K. Image Processing- Principles and Applications. John Wiley & Sons, Inc. 2005.
- [12] Forsyth D. and Ponce J. Computer Vision: A modern approach. Prentice Hall. 2001.
- [13] OpenCV. Sobel Derivatives. 2014. Retirado de: [http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/sobel\\_derivatives/sobel\\_derivatives.html#sobel-derivatives](http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/sobel_derivatives/sobel_derivatives.html#sobel-derivatives).
- [14] OpenCV. Canny Edge Detector. 2014. Retirado de: [http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/canny\\_detector/canny\\_detector.html#canny-detector](http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html#canny-detector),

- [15] Davies ER. Machine Vision: Theory, Algorithms, Practabilities, Morgam Kaufmann. Elsevier.2012.
- [16] Foresti GL and Gentili S. A vision based system for object detection In underwater images. International Journal of Pattern Recognition and Artificial Intelligence. 2000, 167-188.
- [17] Caetano M. Filtro de Kalman: Teoria e Aplicação para Iniciantes. Sd. Retirado de: [http://www.mudancasabruptas.com.br/Filtro%20de%20Kalman\\_Iniciantes.pdf](http://www.mudancasabruptas.com.br/Filtro%20de%20Kalman_Iniciantes.pdf)
- [18] Lopes dos Santos P. “Filtro de kalman”. Texto de apoio à unidade curricular Identificação e Estimação. DEEC, FEUP, Novembro, 2012.
- [19] Grewal MS and Andrews AP. Kalman Filtering: Theory and Practice. Prentice Hall. 1993.
- [20] Ribeiro M. Kalman and Extended Kalman Filters: Concept, Derivation and Properties. Instituto de Sistemas e Robótica, Instituto Superior Técnico. 2004.
- [21] OceanSys. Technology - Systems and Prototypes. Retirado de: <http://oceansys.fe.up.pt/technology.php>.
- [22] OpenCV. Canny. 2014. Retirado de: [http://docs.opencv.org/modules/imgproc/doc/feature\\_detection.html?highlight=canny#canny](http://docs.opencv.org/modules/imgproc/doc/feature_detection.html?highlight=canny#canny).
- [23] OpenCV. findContours. 2014. Retirado de: [http://docs.opencv.org/modules/imgproc/doc/structural\\_analysis\\_and\\_shape\\_descriptors.html?highlight=findcontours#findcontours](http://docs.opencv.org/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html?highlight=findcontours#findcontours).
- [24] OpenCV. contourArea. 2014. Retirado de: [http://docs.opencv.org/modules/imgproc/doc/structural\\_analysis\\_and\\_shape\\_descriptors.html#contourarea](http://docs.opencv.org/modules/imgproc/doc/structural_analysis_and_shape_descriptors.html#contourarea),
- [25] Melo J. and Matos A. Guidance and control of an ASV in AUV tracking operations. 2008.